Sprint

Charles S. "Steve" Parrott
Director - State Regulatory Affairs

Mid-Atlantic Operations
14111 Capital Boulevard
Wake Forest, North Carolina 27587-5900
Telephone: (919) 554-7039
Fax: (919) 554-7595

November 14, 1997

Dr. Austin Lyons
Telecommunications
Tennessee Regulatory Authority
460 James Robertson Parkway
Nashville, TN 37243-0505

RE: Docket No. 97-00888 (Universal Service)
Response to Issue No. 16

Dear Dr. Lyons:

Enclosed is the response of United Telephone-Southeast, Inc. to Issue No. 16, Phase II - Cost Issues, of the Tennessee Regulatory Authority's List of Issues as set forth in the TRA's conference of October 30, 1997. Please contact me or Laura Sykora at 919-554-7323 if you have questions regarding this information.

Sincerely yours,

C. Steve Parrott

Enclosures

c: Laura Sykora
Jim Wright
Bob Wallace
Counsel of Record

CERTIFICATE
Universal Service Fund
(Docket No. 97-00888)

The undersigned hereby certifies that a copy of the Response of United Telephone-Southeast, Inc. to Issue No. 16 filed in the above docket is being provided to each of the following, by placing a copy of the same in the United States Mail postage prepaid and addressed as follows:

Guy M. Hicks
BellSouth Telecommunications, Inc.
333 Commerce St., Suite 2101
Nashville, TN 37201-3300

William Ellenberg/ Bennett Ross
BellSouth Telecommunications, Inc.
Suite 4300
675 West Peachtree Street, NE
Atlanta, Georgia 30375

T. G. Pappas/ John Hayworth
Attorneys for Coalition of Small LECs
Bass, Berry & Sims
2700 First American Center
Nashville, TN 37238-2700

Carolyn Tatum Roddy
Sprint Communications Co. L.P.
3100 Cumberland Circle
Atlanta, GA 30339

Richard M. Tettlebaum
Citizens Telecom
Suite 500
1400 16th Street NW
Washington, DC 20036

Jon E. Hastings
Attorney for MCI
Boult, Cummings, Conners & Berry
414 Union St., Suite 1600
Nashville, TN 37219

Vincent Williams
Consumer Advocate Division
404 James Robertson Parkway
Nashville, TN 37219

Charles B. Welch/Tony Thompson
Attorneys for Time Warner
Farris, Mathews, Gilman
511 Union St., Suite 2400
Nashville, TN 37219

James Lamoureux
AT&T
Room 4068
1200 Peachtree Street, N.E.
Atlanta, Ga. 30309

Val Sanford
Attorney for AT&T
Gullett, Sanford, Robinson & Martin
230 Fourth Avenue North, 3rd Floor
Nashville, TN 37219-8888

Henry Walker
Attorney for NextLink and ACSI
Boult Cummings, Conners & Berry
414 Union Street, Suite 1600
Nashville, TN 37219

Fred L. Terry, General Manager
Highland Telephone Cooperative Inc.
P. O. Box 119
Sunbright, TN 37872

Pam Melton
Attorney for LCI
8180 Greensboro Drive
Suite 800
McLean, VA 22102

Wayne Gasgaway
Manager
DeKalb Telephone Cooperative
P. O. Box 347
Alexandria, TN 37012

Glen B. Sears, General Manager
West Kentucky Rural Telephone Cooperative
237 North 8th Street
Mayfield, KY 42066

D. Billye Sanders
Attorney for TCG MidSouth
P. O. Box 198966
Nashville, TN 37219-8966

H. LaDon Baltimore (Attorney for WorldCom)
Suite 320
211 Seventh Avenue N
Nashville, TN 37219-1823

F. Thomas Rowland
North Central Telephone Coop.
P. O. Box 70
Lafayette, TN 37083-0070

W. T. Sims
Manager
Yorkville Telephone Cooperative
Yorkville, TN 38389

Richard Cys (Attorney for NextLink)
Davis Wright Tremaine
1155 Connecticut Avenue NW
Suite 700
Washington, DC 20036

Richard Smith, President
Standard Communications Co.
302 Sunset Drive
Suite 101
Johnson City, TN 37604

James W. Dempster
(Attorney for Ben Lomand Rural Telephone Coop.)
P. O. Box 332
McMinnville, TN 37111-0333

Daniel M. Waggoner (Attorney for NextLink)
Davis Wright Tremaine
2600 Century Square
1501 Fourth Avenue
Seattle, WA 98101-1685

Thomas J. Curran
Director External Affairs
360 Communications Company
8725 W. Higgins Road
Chicago, IL 60631

Nanette Edwards
Regulatory Affairs Manager
DeltaCom
700 Blvd. South, Suite 101
Hunteville, AL 35801

Ellen Bryson
Bledsoe Telephone Coop.
5010 Hodgkins Plaza
Dilburn,GA 30047

Proctor Upchurch
Attorney for Bledsoe Telephone Coop.
P. O. Box 3549
Woodmere Mall
Crossville, TN 38557-3549

Jack McFadden
Department of Finance & Administration
Telecommunications Policy & Planning
598 James Robertson Parkway
Nashville, TN 37243-0560

Bryan McCarty
Attorney for TEA Legal Services
501 Second Avenue N
Nashville, TN 37201-1099

Joe Reeves
WorldCom and LCI
Suite 320
211 Seventh Avenue N
Nashville, TN 37219

Gif Thornton
Attorney for BellSouth Cellular
424 Church Street
28th Floor
Nashville, TN 37219-2380

Sheila Davis
Chaz Taylor Inc.
3401 West End Avenue
Suite 318
Nashville, TN 37203

Robert D. Dudney
General Manager
Twin Lakes Telephone Coorperative Corp.
P. O. Box 67
Gainesboro, TN 38562

Mark Pasko/ Michael Romano
Swidler & Berlin
Atty. for AVR d/b/a Hyperion of TN
3000 K Street NW, Suite 300
Washington, DC 20007-5116

Dr. Peggy Smith
Chair. Tennessee EdLine
801 Second Avenue N
Nashville, TN 37201

Dana Shaffer
NextLink Tennessee
105 Molloy Street
Suite 300
Nashville, TN 37201

Phillip Carver
BellSouth Telecommunications
675 West Peachtree Street
Atlanta, GA 30375

William C. Carriger
Atty. for Electric Power Bd. of Chattanooga
400 Keyatel Building
One Union Square
Chattanooga, TN 37402

Denise Newman
Phoenix Network Inc.
1687 Cola Blvd.
Bolden, CO 80401

Jane Walters
TN Dept. of Education
6th Floor, Gateway Plaza
710 James Robertson Parkway
Nashville, TN 37243-0375

Dan Elrod & Ken Bryant
Attorneys for GTE Mobilnet
Nashville City Center, 25th Floor
511 Union Street
Nashville, TN 37219

Ozle Allen
Tennessee Co-Ops
5755 Short Mountain Road
McMinnville, TN 37110

Archie Hickerson
Consumer Advocate
2nd Floor, Cordell Hull Bldg.
425 Fifth Avenue North
Nashville, Tn 37219

This 14th day of November, 1997 James B Wright #12805

UNIVERSAL SERVICE, DOCKET NO. 97-00888
PHASE II - COST ISSUES
RESPONSE TO ISSUE NO. 16

Question 16 What cost model or method should be adopted to calculate needed universal service supports? **Note: the word "method" is used to mean "algorithm(s) and input value(s)."**

Response: The Benchmark Cost Proxy Model (BCPM) Release 3.0 should be adopted to calculate universal service supports. The Model Methodology is attached to and referenced in the responses to the questions 16a-y.

UNIVERSAL SERVICE, DOCKET NO. 97-00888
PHASE II - COST ISSUES
RESPONSE TO ISSUE NO. 16

Question 16 (a) What method should be used to population distributions within service areas?

Response:The best method to use for population distribution (or customer location) is one which:

- does NOT make ad hoc assumptions regarding where customers might be clustered together
- does use publicly available data disaggregated down to the most granular level possible
- does (only *after* the above-mentioned disaggregation) make reasonable and valid assumptions regarding relative customer location within this granular level of detail.

This is the method used by the BCPM3.0, outlined in the document Model Methodology (attached) on pages 23-31. The BCPM3.0 uses publicly available census data which identifies customer locations down to the census block level (a much finer level of granularity than the census block group. A single census block group usually contains many census blocks.) After identifying the number of customers located in each census block, the model overlays this area with a grid network to identify portions of land area that contain no roads (and thereby are less likely to contain customer locations. This assumption has been validated by numerous econometric tests.) Next, known counts of customer locations are distributed along road segments which allows for any clustering of customers that actually exists to be *revealed* through the location process, instead of clusters being *invented* by the location process.

For example, picture two separate census blocks A and B, each 5 square miles in area and each holding 100 customers. Assume Census Block A only has roads in a small portion of its southwest corner, while CensusBlock B has roads criss-crossing the entire 5 mile area. It is reasonable to model the customers in Census Block A as more "clustered" than those in Census Block B. The BCPM3.0 will do this. Other proxy models will not.

UNIVERSAL SERVICE, DOCKET NO. 97-00888
PHASE II - COST ISSUES
RESPONSE TO ISSUE NO. 16

Question 16 (b) What method should be used to match a model's wire center line count to a LEC's existing wire center line count?

Response: The model should be able to use the LEC's actual line count instead of the estimated line count when it builds the network and calculates the costs. The BCPM3.0 has this capability. If actual wire center line counts are available, the model can incorporate that specific number into its outside plant algorithm to determine the size and scope of the network that must be constructed.

A key point regarding this issue is that the accurate line counts must be distributed over the accurate land area. It does no good to match a LEC's line counts if the distance those lines must cover from wire center to customer is misstated. Any model that uses census block groups (CBGs) as a basic unit of analysis will misstate costs because CBG boundaries often have no relationship at all to wire center boundaries.

Many CBGs are actually served by multiple wire centers, sometimes from completely different phone companies. If such a CBG is assigned to a single wire center, the land area assigned to that wire center will be overstated and the land areas served by the adjoining wire centers will be understated (since some of their land area will be mistakenly "assigned" to the first wire center). "Truing up" line counts, without truing up the area that those lines cover, distorts distance and density measures which are the key drivers of cost.

UNIVERSAL SERVICE, DOCKET NO. 97-00888
PHASE II - COST ISSUES
RESPONSE TO ISSUE NO. 16

Question 16 (c) What method should be used to determine the proper outside plant mix (i.e., the fractions of aerial, underground, and buried cable) and associated materials and installation costs?

Response: At a very minimum, outside plant mix should be allowed to vary 1) between feeder and distribution segments, 2) with the relative density of the area being served, 3) depending on the type of material used (copper/fiber), and 4) depending on the terrain of the area (normal, soft rock, hard rock, etc.). The BCPM3.0 allows for this variation as described in the Model Methodology document.

The actual mix percentages should be calculated by individuals familiar with both standard engineering practices and the specific area in question, keeping in mind the forward-looking nature of the model. Most probably, these would be the engineers responsible for outside plant design associated with incumbent LECs serving the actual area.

UNIVERSAL SERVICE, DOCKET NO. 97-00888
PHASE II - COST ISSUES
RESPONSE TO ISSUE NO. 16

Question 16 (d) What method should be used to determine drop lengths and associated costs?

Response: Drop lengths and costs should be determined as they are in the BCPM3.0 and as they are documented in the attached Excel document. This Excel document presents the actual calculation algorithms, cell references and equations used in the BCPM3.0 to determine drop length and associated cost.

Worksheets:
Grid Data
Grid Demographics
Distance Calculations
DLC to FDI
New Distribution
Term,Drop,&NID
Subfeeder Part2
SubFeeder
Main Feeder
Electronics&FDI
Main&SubfeederAllocation
Switching
Investment
Output
Miscellaneous Inputs
Electronic Tables
Miscellaneous Tables
Unit By Density Tables
Density By Unit Tables
Fixed Tables
FCC Data

Sheet: Grid Data

Col	Range Name	Column Name	Column Comment	Formula	Formula Comment
A	CLLI*	Clli			
B	Latitude	Latitude			
C	Longitude	Longitude			
D	Area_sq_Miles	Area-sq Miles			
E	DepthtoBedrock	Depth to Bedrock (Inches)			
F	RockHardness	Rock Hardness			
G	SurfaceSoilTexture	Surface Soil Texture			
H	WaterTableDepth	Water Table Depth (Feet)			
I	MinimumSoilSlope	Minimum Soil Slope			
J	MaximumSoilSlop e	Maximum Soil Slope			
K	TotalBusinessLines	Total Business Lines			
L	BusinessLocations	Business Locations			
M	TotalHouseholds	Total Households			
N	HousingUnits	Housing Units			

O	SingleUnitDetachedHUlocations	Single Unit Detached HU Locations			
P	SingleUnitAttachedHUlocations	Single Unit Attached HU Locations			
Q	_2HUPerDwellinglocations	2 HU Per Dwelling Locations			
R	_3to4HUperdwellinglocations	3-4 HU Per Dwelling Locations			
S	_5to9HUperdwellinglocations	5-9 HU Per Dwelling Locations			
T	_10to25HUperdwellinglocations	10-19 HU Per Dwelling Locations			
U	_25to50HUperdwellinglocations	20-49 HU Per Dwelling Locations		= IF (ActualLoopCostPerLine > InvLoopCap, InvLoopCap, ActualLoopCostPerLine)	
V	Over50HUperdwellinglocations	>50 HU Per Dwelling Locations			
W	MobileHUlocations	Mobile HU Locations			
X	OtherHUlocations	Other HU Locations			
Y	LatitudeAtRoadCentroid	Latitude At Road Centroid			
Z	LongitudeAtRoadCrossing	Longitude At Road Centroid			
AA	AirlineDistance	Airline Distance Feet			
AB	FDICode*	FDI Code			
AC	MainFeederLength	Main Feeder Length			
AD	SubfeederLength	Sub Feeder Length			
AE	Part2SubfeederLength	Part 2 SubFeeder Length			
AF	ULQHousingUnits	ULQ Housing Units			
AG	ULQHouseholds	ULQ Households			
AH	ULQBusinessLines	ULQ Business Lines			
AI	ULQRoadLength	ULQ Road Length			
AJ	ULQArea	ULQ Road Reduced Area			
AK	PreULQHorizontalfttoRoadCentroid	Preliminary ULQ Horizontal Feet To Road Centroid			
AL	PreULQverticalfttoRoadCentroid	Preliminary ULQ Vertical Feet To Road Centroid			
AM	URQHousingUnits	URQ Housing Units			
AN	URQHouseholds	URQ Households			

AO	URQBusinessLines	URQ Business Lines			
AP	URQRoadLength	URQ Road Length			
AQ	URQArea	URQ Road Reduced Area			
AR	PreURQHorizontalfttoRoadCentroid	Preliminary URQ Horizontal Feet To Road Centroid			
AS	PreURQverticalfttoRoadCentroid	Preliminary URQ Vertical Feet To Road Centroid			
AT	LLQHousingUnits	LLQ Housing Units			
AU	LLQHouseholds	LLQ Households			
AV	LLQBusinessLines	LLQ Business Lines			
AW	LLQRoadLength	LLQ Road Length			
AX	LLQArea	LLQ Road Reduced Area			
AY	PreLLQHorizontalfttoRoadCentroid	Preliminary LLQ Horizontal Feet To Road Centroid			
AZ	PreLLQverticalfttoRoadCentroid	Preliminary LLQ Vertical Feet To Road Centroid			
BA	LRQHousingUnits	LRQ Housing Units			
BB	LRQHouseholds	LRQ Households			
BC	LRQBusinessLines	LRQ Business Lines			
BD	LRQRoadLength	LRQ Road Length			
BE	LRQArea	LRQ Road Reduced Area			
BF	PreLRQHorizontalfttoRoadCentroid	Preliminary LRQ Horizontal Feet To Road Centroid			
BG	PreLRQverticalfttoRoadCentroid	Preliminary LRQ Vertical Feet To Road Centroid			

Sheet: Grid Demographics

Col	Range Name	Column Name	Column Comment	Formula	Formula Comment
A	CLLIQuad*	CLLIQUAD		= LEFT (FDICode, 1)	
B	CLLISegment*	CLLISEGMENT		= LEFT (FDICode, 2)	
C	MainFeederBranch *	Main Feeder Branch Number		= MID (FDICode, 2, 1)	
D	StateId	State Id		= MID (CLLI, 5, 2)	
E	CompanySize	Company Size	Company Size: 1 = Small 2 = Medium 3 = Large	= VLOOKUP (CLLI, CompanySizeTable, 2)	
F	FeederSegmentSequenceNumber*	Feeder Segment Sequence Number		= IF (CLLIQuad = PrevCLLIQuad, PrevFeederSegmentSequenceNumber + 1, 1)	
G	MainFeederSequenceNumber*	Main Feeder Sequence Number		= IF (AND (CLLIQuad = PrevCLLIQuad, CLLISegment = PrevCLLISegment) , PrevMainFeederSequenceNumber + 1, 1)	
H	SingleResidenceLines	Single Residence Lines		= TotalHouseholds	
I	AdditionalResidenceLines	Additional Residence Lines		= TotalHouseholds * (VLOOKUP (StateId, StateInformationTable, 2) - 1)	
J	ResidenceLineBalancingFctr	Residence Line Balancing Factor		= IF (ISNA (VLOOKUP (LEFT (CLLI, 8) , FCCLineTable, 2, FALSE)) , 1, VLOOKUP (LEFT (CLLI, 8) , FCCLineTable, 2, FALSE) / (SUM (TotalHousehoIdsColumn) + SUM (AdditionalResidenceLinesColumn)))	
K	ResidenceLines	Total Residence Lines		= ResidenceLineBalancingFctr * (SingleResidenceLines + AdditionalResidenceLines)	
L	SingleBusinessLines	Single Business Lines		= IF (ISNA (VLOOKUP (LEFT (CLLI, 8) , FCCLineTable, 3, FALSE)) , TotalBusinessLines * VLOOKUP (StateId, StateInformationTable, 3) , TotalBusinessLines * IF (SUM (TotalBusinessLinesColumn) = 0, 0, VLOOKUP (LEFT (CLLI, 8) , FCCLineTable, 3, FALSE) / SUM (TotalBusinessLinesColumn)))	
M	MultiBusinessLines	Multiple Business Lines		= IF (ISNA (VLOOKUP (LEFT (CLLI, 8) , FCCLineTable, 4, FALSE)) , TotalBusinessLines * (1 - VLOOKUP (StateId, StateInformationTable, 3)) , TotalBusinessLines * IF (SUM (TotalBusinessLinesColumn) = 0, 0, VLOOKUP (LEFT (CLLI, 8) , FCCLineTable, 4, FALSE) / SUM (TotalBusinessLinesColumn)))	
N	BusinessLineBalancingFctr	Business Line Balancing Factor		= IF (TotalBusinessLines = 0, 0, (SingleBusinessLines + MultiBusinessLines) / TotalBusinessLines)	
O	SpecialAccessLines	Special Access Lines		= IF (ISNA (VLOOKUP (LEFT (CLLI, 8) , FCCLineTable, 5, FALSE)) , TotalBusinessLines * VLOOKUP (StateId, StateInformationTable, 4) , TotalBusinessLines * IF (SUM (TotalBusinessLinesColumn) = 0, 0, VLOOKUP (LEFT (CLLI, 8) , FCCLineTable, 5, FALSE) / SUM (TotalBusinessLinesColumn)))	
P	TotalWorkingLinesServedInGrid	Total Working Lines Served In Grid		= ResidenceLines + SingleBusinessLines + MultiBusinessLines + SpecialAccessLines	
Q	TotalLinesServedInGrid	Total Lines Served In Grid		= IF ((HousingUnits - TotalHouseholds) > 0, TotalWorkingLinesServedInGrid + (HousingUnits - TotalHouseholds) , TotalWorkingLinesServedInGrid)	
R	Density*	Density in Lines/SQ MI		= IF (Area_sq_Miles = 0, IF (ISNUMBER (PrevDensity) , PrevDensity, NextDensity) , TotalLinesServedInGrid / Area_sq_Miles)	
S	DensityColumnPointer	Density Column Pointer		= VLOOKUP (Density, DensityPointerTable, 2)	
T	GridLinesProvisionedAsDS1s	Grid Lines Provisioned as DS-1s		= SpecialAccessLines * VLOOKUP (TotalLinesServedInGrid, VoiceGradeRatioTable, 5) + SwitchedLinesInGrid * VLOOKUP (TotalLinesServedInGrid, VoiceGradeRatioTable, 3)	
U	SwitchedLinesInGrid	Switched Lines in Grid		= ResidenceLines + SingleBusinessLines + MultiBusinessLines	
V	SoilSlopeFactor	Soil Slope Factor		= IF (MinimumSoilSlope > MinSlopeTrigger, IF (MaximumSoilSlope > MaxSlopeTrigger, CombSlopeFactor, MinSlopeFactor) , IF (MaximumSoilSlope > MaxSlopeTrigger, MaxSlopeFactor, 1))	

W	CopperDepthCondition	Copper Depth Condition	RockH = 1 RockS = 2 Normal = 3	= IF (AND (DepthtoBedrock < = NormalUGBuriedCover, RockHardness = "HARD") , 1, IF (AND (DepthtoBedrock > NormalUGBuriedCover, SurfaceIndicator = 0) , 3, 2))	
X	FiberDepthCondition	Fiber Depth Condition	RockH = 1 RockS = 2 Normal = 3	= IF (AND (DepthtoBedrock < = NormalFiberCover, RockHardness = "HARD") , 1, IF (AND (DepthtoBedrock > NormalFiberCover, SurfaceIndicator = 0) , 3, 2))	
Y	WaterCostMultiplier	Water Cost Multiplier		= IF (WaterTableDepth < CriticalWaterDepth, 1 + (WaterFactor / 100) , 1)	
Z	SurfaceIndicator	Surface Indicator		= IF (ISBLANK (SurfaceSoilTexture) , 0, IF (ISERROR (VLOOKUP (SurfaceSoilTexture, SurfaceTextureTable, 2)) = TRUE, 0, VLOOKUP (SurfaceSoilTexture, SurfaceTextureTable, 2)))	
AA	_1to2Buildings	2 HU Per Buildings		= _2HUPerDwellinglocations / (VLOOKUP ("2", HousingUnitsperDwelling, DensityColumnPointer, FALSE))	
AB	_3to4Buildings	3-4 HU Per Buildings		= _3to4HUperdwellinglocations / (VLOOKUP ("3 - 4", HousingUnitsperDwelling, DensityColumnPointer, FALSE))	
AC	_5to9Buildings	5-9 HU Per Buildings		= _5to9HUperdwellinglocations / (VLOOKUP ("5 - 9", HousingUnitsperDwelling, DensityColumnPointer, FALSE))	
AD	_10to19Buildings	10-19 HU Per Buildings		= _10to25HUperdwellinglocations / (VLOOKUP ("10 - 19", HousingUnitsperDwelling, DensityColumnPointer, FALSE))	
AE	_20-49Buildings	20-49 HU Per Buildings		= _25to50HUperdwellinglocations / (VLOOKUP ("20 - 49", HousingUnitsperDwelling, DensityColumnPointer, FALSE))	
AF	Over50Buildings	>50 HU Per Buildings		= Over50HUPerDwellingLocations / (VLOOKUP (" > 50", HousingUnitsperDwelling, DensityColumnPointer, FALSE))	
AG	OtherBuildings	Other HU Buildings		= OtherHULocations / (VLOOKUP ("Other", HousingUnitsperDwelling, DensityColumnPointer, FALSE))	

Sheet: Distance Calculations

Col	Range Name	Column Name	Column Comment	Formula	Formula Comment
A	TotalHousingUnit Locations	Total Housing Unit Locations		= SingleUnitDetachedHUlocations + SingleUnitAttachedHUlocations + _1to2Buildings + _3to4Buildings + _5to9Buildings + _10to19Buildings + _20to49Buildings + over50Buildings + MobileHUlocations + otherBuildings	
B		Total Business Locations		= BusinessLocations	
C	ULQHousingUnitslocations	ULQ Housing Units Locations		= IF (HousingUnits = 0, 0, TotalHousingUnitLocations * ULQHousingUnits / HousingUnits)	
D	ULQBusinesslocations	ULQ Business Locations		= IF (TotalBusinessLines = 0, 0, CEILING (BusinessLocations * ULQBusinessLines / TotalBusinessLines, 1))	
E	ULQLots	ULQ Lots		= ULQHousingUnitslocations + ULQBusinesslocations	
F	ULQLotSideLength	ULQ Lot Side Length		= IF (ULQLots = 0, 0, SQRT (ULQArea / ULQLots) * 5280)	
G	ULQQuadrantSide Length	ULQ Quadrant Side Length		= SQRT (ULQArea) * 5280	
H	PreULQBackboneandBranchCableLength	Preliminary ULQ Backbone & Branch Cable Length		= IF (NumberpairsinULquadrant = 0, 0, IF (ULQQuadrantSideLength - (2 * ULQLotSideLength) < 0, 0, ULQQuadrantSideLength - (2 * ULQLotSideLength)))	
I	NumberULQbranchcables	Number Of ULQ Branch Cables		= IF (OR (ULQLotSideLength = 0, PreULQBackboneandBranchCableLength = 0) , 0, CEILING (ULQQuadrantSideLength / (2 * ULQLotSideLength) , 1))	
J	ULQRoadLengthAdj	ULQ Road Length Adjustment		= IF (ULQRoadLength = 0, 1, IF (((NumberULQbranchcables + 1 / 2) * PreULQBackboneandBranchCableLength + PreULQHorizontalfttoRoadCentroid + PreULQverticalfttoRoadCentroid) - ULQRoadLength > 0, ((ULQRoadLength / ((NumberULQbranchcables + 1 / 2) * PreULQBackboneandBranchCableLength + PreULQHorizontalfttoRoadCentroid + PreULQverticalfttoRoadCentroid))) , 1))	
K	ULQBackboneandBranchCableLength	ULQ Backbone &Branch Cable Length (Road Adjusted)		= PreULQBackboneandBranchCableLength * ULQRoadLengthAdj	
L	NumberpairsinULquadrant	Number Of Pairs In UL Quadrant		= ULQHousingUnits * PairsPerHousingUnit + IF (ULQBusinesslocations = 0, 0, IF ((ULQBusinessLines / ULQBusinesslocations) > PairsPerBusinessLocation, ULQBusinessLines, PairsPerBusinessLocation * ULQBusinesslocations))	
M	LLQHousingUnitlocations	LLQ Housing Unit Locations		= IF (HousingUnits = 0, 0, TotalHousingUnitLocations * LLQHousingUnits / HousingUnits)	
N	LLQBusinesslocations	LLQ Business Locations		= IF (TotalBusinessLines = 0, 0, CEILING (BusinessLocations * LLQBusinessLines / TotalBusinessLines, 1))	
O	LLQLots	LLQ Lots		= LLQHousingUnitlocations + LLQBusinesslocations	
P	LLQLotSideLength	LLQ Lot Side Length		= IF (LLQLots = 0, 0, SQRT (LLQArea / LLQLots) * 5280)	
Q	LLQQuadrantSide Length	LLQ Quadrant Side Length		= SQRT (LLQArea) * 5280	
R	PreLLQBackboneBranchCableLength	Preliminary LLQ Backbone & Branch Cable Length		= IF (NumberpairsinLLquadrant = 0, 0, IF (LLQQuadrantSideLength - (2 * LLQLotSideLength) < 0, 0, LLQQuadrantSideLength - (2 * LLQLotSideLength)))	
S	NumberLLQbranchcables	Number Of LLQ Branch Cables		= IF (OR (LLQLotSideLength = 0, PreLLQBackboneBranchCableLength = 0) , 0, CEILING (LLQQuadrantSideLength / (2 * LLQLotSideLength) , 1))	
T	LLQRoadLengthAdj	LLQ Road Length Adjustment		= IF (LLQRoadLength = 0, 1, IF (((NumberLLQbranchcables + 1 / 2) * PreLLQBackboneBranchCableLength + PreLLQHorizontalfttoRoadCentroid + PreLLQVerticalFtToRoadCentroid) - LLQRoadLength > 0, ((LLQRoadLength / ((NumberLLQbranchcables + 1 / 2) * PreLLQBackboneBranchCableLength + PreLLQHorizontalfttoRoadCentroid + PreLLQVerticalFtToRoadCentroid))) , 1))	

U	LLQBackboneBranchCableLength	LLQ Backbone &Branch Cable Length (Road Adjusted)		= PreLLQBackboneBranchCableLength * LLQRoadLengthAdj	
V	NumberpairsinLLquadrant	Number Of Pairs In LL Quadrant		= LLQHousingUnits * PairsPerHousingUnit + IF (LLQBusinesslocations = 0, 0, IF ((LLQBusinessLines / LLQBusinesslocations) > PairsPerBusinessLocation, LLQBusinessLines, PairsPerBusinessLocation * LLQBusinesslocations))	
W	URQHousingUnitlocations	URQ Housing Unit Locations		= IF (HousingUnits = 0, 0, TotalHousingUnitLocations * URQHousingUnits / HousingUnits)	
X	URQBusinesslocations	URQ Business Locations		= IF (TotalBusinessLines = 0, 0, CEILING (BusinessLocations * URQBusinessLines / TotalBusinessLines, 1))	
Y	URQLots	URQ Lots		= URQHousingUnitlocations + URQBusinesslocations	
Z	URQLotSideLength	URQ Lot Side Length		= IF (URQLots = 0, 0, SQRT (URQArea / URQLots) * 5280)	
AA	URQQuadrantSideLength	URQ Quadrant Side Length		= SQRT (URQArea) * 5280	
AB	PreURQBackboneBranchCableLength	Preliminary URQ Backbone & Branch Cable Length		= IF (NumberpairsinURquadrant = 0, 0, IF (URQQuadrantSideLength - (2 * URQLotSideLength) < 0, 0, URQQuadrantSideLength - (2 * URQLotSideLength)))	
AC	NumberURQbranchcables	Number Of URQ Branch Cables		= IF (OR (URQLotSideLength = 0, PreURQBackboneBranchCableLength = 0) , 0, CEILING (URQQuadrantSideLength / (2 * URQLotSideLength) , 1))	
AD	URQRoadLengthAdj	URQ Road Length Adjustment		= IF (URQRoadLength = 0, 1, IF (((NumberURQbranchcables + 1 / 2) * PreURQBackboneBranchCableLength + PreURQHorizontalfttoRoadCentroid + PreURQverticalfttoRoadCentroid) - URQRoadLength > 0, ((URQRoadLength / ((NumberURQbranchcables + 1 / 2) * PreURQBackboneBranchCableLength + PreURQHorizontalfttoRoadCentroid + PreURQverticalfttoRoadCentroid))) , 1))	
AE	URQBackboneBranchCableLength	URQ Backbone &Branch Cable Length (Road Adjusted)		= PreURQBackboneBranchCableLength * URQRoadLengthAdj	
AF	NumberpairsinURquadrant	Number Of Pairs In UR Quadrant		= URQHousingUnits * PairsPerHousingUnit + IF (URQBusinesslocations = 0, 0, IF ((URQBusinessLines / URQBusinesslocations) > PairsPerBusinessLocation, URQBusinessLines, PairsPerBusinessLocation * URQBusinesslocations))	
AG	LRQHousingUnitlocations	LRQ Housing Unit Locations		= IF (HousingUnits = 0, 0, TotalHousingUnitLocations * LRQHousingUnits / HousingUnits)	
AH	LRQBusinesslocations	LRQ Business Locations		= IF (TotalBusinessLines = 0, 0, CEILING (BusinessLocations * LRQBusinessLines / TotalBusinessLines, 1))	
AI	LRQLots	LRQ Lots		= LRQHousingUnitlocations + LRQBusinesslocations	
AJ	LRQLotSideLength	LRQ Lot Side Length		= IF (LRQLots = 0, 0, SQRT (LRQArea / LRQLots) * 5280)	
AK	LRQQuadrantSideLength	LRQ Quadrant Side Length		= SQRT (LRQArea) * 5280	
AL	PreLRQBackboneBranchCableLength	Preliminary LRQ Backbone & Branch Cable Length		= IF (NumberpairsinLRquadrant = 0, 0, IF (LRQQuadrantSideLength - (2 * LRQLotSideLength) < 0, 0, LRQQuadrantSideLength - (2 * LRQLotSideLength)))	
AM	NumberLRQbranchcables	Number Of LRQ Branch Cables		= IF (OR (LRQLotSideLength = 0, PreLRQBackboneBranchCableLength = 0) , 0, CEILING (LRQQuadrantSideLength / (2 * LRQLotSideLength) , 1))	
AN	LRQRoadLengthAdj	LRQ Road Length Adjustment		= IF (LRQRoadLength = 0, 1, IF (((NumberLRQbranchcables + 1 / 2) * PreLRQBackboneBranchCableLength + PreLRQHorizontalfttoRoadCentroid + PreLRQVerticalFtToRoadCentroid) - LRQRoadLength > 0, ((LRQRoadLength / ((NumberLRQbranchcables + 1 / 2) * PreLRQBackboneBranchCableLength + PreLRQHorizontalfttoRoadCentroid + PreLRQVerticalFtToRoadCentroid))) , 1))	

AO	LRQBackboneBranchCableLength	LRQ Backbone &Branch Cable Length (Road Adjusted)		= PreLRQBackboneBranchCableLength * LRQRoadLengthAdj	
AP	NumberpairsinLRquadrant	Number Of Pairs In LR Quadrant		= LRQHousingUnits * PairsPerHousingUnit + IF (LRQBusinesslocations = 0, 0, IF ((LRQBusinessLines / LRQBusinesslocations) > PairsPerBusinessLocation, LRQBusinessLines, PairsPerBusinessLocation * LRQBusinesslocations))	
AQ	ULQHorizontalfttoRoadCentroid	ULQ Horizontal Feet To Road Centroid (Road Adjusted)		= PreULQHorizontalfttoRoadCentroid * ULQRoadLengthAdj	
AR	ULQverticalfttoRoadCentroid	ULQ Vertical Feet To Road Centroid (Road Adjusted)		= PreULQverticalfttoRoadCentroid * ULQRoadLengthAdj	
AS	URQHorizontalfttoRoadCentroid	URQ Horizontal Feet To Road Centroid (Road Adjusted)		= PreURQHorizontalfttoRoadCentroid * URQRoadLengthAdj	
AT	URQverticalfttoRoadCentroid	URQ Vertical Feet To Road Centroid (Road Adjusted)		= PreURQverticalfttoRoadCentroid * URQRoadLengthAdj	
AU	LLQHorizontalfttoRoadCentroid	LLQ Horizontal Feet To Road Centroid (Road Adjusted)		= PreLLQHorizontalfttoRoadCentroid * LLQRoadLengthAdj	
AV	LLQverticalfttoRoadCentroid	LLQ Vertical Feet To Road Centroid (Road Adjusted)		= PreLLQVerticalFtToRoadCentroid * LLQRoadLengthAdj	
AW	LRQHorizontalfttoRoadCentroid	LRQ Horizontal Feet To Road Centroid (Road Adjusted)		= PreLRQHorizontalfttoRoadCentroid * LRQRoadLengthAdj	
AX	LRQverticalfttoRoadCentroid	LRQ Vertical Feet To Road Centroid (Road Adjusted)		= PreLRQVerticalFtToRoadCentroid * LRQRoadLengthAdj	
AY	MainFeederDistance*	Main Feeder Distance		= MainFeederLength	
AZ	PrimarySubfeederDistance	Primary Subfeeder Distance		= SubfeederLength	
BA	SubfeederPart2Distance	Subfeeder Part 2 Distance		= Part2SubfeederLength	
BB	MaxDLCTOFDICableDistance	Maximum DLC to FDI Cable Distance		= MAX (ULQHorizontalfttoRoadCentroid + ULQverticalfttoRoadCentroid, LLQHorizontalfttoRoadCentroid + LLQverticalfttoRoadCentroid, URQHorizontalfttoRoadCentroid + URQverticalfttoRoadCentroid, LRQHorizontalfttoRoadCentroid + LRQverticalfttoRoadCentroid)	
BC	TotalFeederPlanningDistance	Total Feeder Planning Distance		= SoilSlopeFactor * (MainFeederDistance + PrimarySubfeederDistance + SubfeederPart2Distance + MaxDLCTOFDICableDistance)	
BD	GridSideLength	Grid Side Length		= (SQRT (CHOOSE (LargestQuadAreaNumber, ULQArea, LLQArea, LRQArea, URQArea)) * 5280) * SoilSlopeFactor	
BE	NumberOfLotsInGrid	Number of lots in Grid		= CHOOSE (LargestQuadAreaNumber, ULQHousingUnits + ULQBusinesslocations, LLQHousingUnits + LLQBusinesslocations, LRQHousingUnits + LRQBusinesslocations, URQHousingUnits + URQBusinesslocations)	
BF	NumberOfLotsPerBaseSide	Number of Lots per Base Side		= MAX (2, CEILING (SQRT (NumberOfLotsInGrid) , 1))	

BG	BaseSideLotLength	Base Side Lot Length		= GridSideLength / NumberOfLotsPerBaseSide	
BH	SecondSideLotLength	Second Side Lot Length		= IF (SQRT (NumberOfLotsInGrid) = 0, 0, BaseSideLotLength * CEILING (SQRT (NumberOfLotsInGrid) , 1) / SQRT (NumberOfLotsInGrid))	
BI	MaxDistributionPlanningDistance	Maximum Distribution Planning Distance		= SoilSlopeFactor * (GridSideLength - BaseSideLotLength - SecondSideLotLength)	
BJ	TotalLoopPlanning Length	Total Loop Planning Length		= TotalFeederPlanningDistance + MaxDistributionPlanningDistance	
BK	FeederFill	Feeder Fill Factor		= VLOOKUP (Density, DensityFillTable, 2)	
BL	DistributionFillFactor	Distribution Fill Factor		= VLOOKUP (Density, DensityFillTable, 3)	
BM	GridPlantType*	Grid Plant Type		= IF (AND (TotalLinesServedInGrid = 0, MainFeederBranch = "0", CLLIQuad = NextCLLIQuad) , NextGridPlantType, IF (AND (TotalLoopPlanningLength < Breakpoint, (NetCopperPairs / FeederFill) < MaxDistrSize) , "Cable", IF (AND (TotalLoopPlanningLength > = Breakpoint, TotalLinesServedInGrid / ElectronicFill < = 240) , "DLC - S", "DLC - L")))	
BN	GridPlantIndex	Grid Plant Index		= IF (GridPlantType = "Cable", 1, IF (GridPlantType = "DLC - S", 2, 3))	
BO	SegmentType1	Segment Type 1	Facilty type that Serves lines in this Grid	= CHOOSE (GridPlantIndex, "Cable", "Fiber", "Fiber")	
BP	SegmentType2	Segment Type 2	Facility Type that is passing through this Grid to Serve Grids farther out	= IF (OR (AND (CLLIQuad = NextCLLIQuad, LEFT (FDICode, 2) = 0) , CLLISegment = NextCLLISegment) , IF (SegmentType1 = NextSegmentType1, NextSegmentType2, NextSegmentType1) , "None")	
BQ	IsSamePlantType	Same Plant Type		= GridPlantType = NextGridPlantType	
BR	NetCopperPairs	Net Grid Pairs if copper		= TotalLinesServedInGrid - GridLinesProvisionedAsDS1s + GridLinesProvisionedAsDS1s / 12	
BS	VoiceGradeLinesEquippedatTerminal	# Voice Grade Lines Equipped at Terminal		= IF (TotalWorkingLinesServedInGrid = 0, 0, CHOOSE (GridPlantIndex, 0, CEILING (TotalWorkingLinesServedInGrid / ElectronicFill, 6) , CEILING ((SwitchedLinesInGrid * VLOOKUP (TotalWorkingLinesServedInGrid, VoiceGradeRatioTable, 2) + SpecialAccessLines * VLOOKUP (TotalWorkingLinesServedInGrid, VoiceGradeRatioTable, 4)) / ElectronicFill, 4)))	
BT	NumberDLCLsatFullCapacity	# DLC-L Systems at Full Capacity		= CHOOSE (GridPlantIndex, 0, 0, TRUNC (VoiceGradeLinesEquippedatTerminal / 1344))	
BU	SwitchedLinesinOverflowDLCLTerminal	# Switched Lines in Overflow DLC-L Terminal		= CHOOSE (GridPlantIndex, 0, 0, VoiceGradeLinesEquippedatTerminal - (1344 * NumberDLCLsatFullCapacity))	
BV	LinesatDS1atTerminal	# DS-1 Lines at Terminal		= IF (TotalLinesServedInGrid = 0, 0, CHOOSE (GridPlantIndex, 0, 0, GridLinesProvisionedAsDS1s / HiCapFill))	
BW	DS3DS1UnitsperTerminalLocation	# DS-3/DS-1 Units per Terminal Location		= CEILING (LinesatDS1atTerminal / (672 * ElectronicFill) , 1)	
BX	NumberFibersPerGrid	Number Fibers per Grid		= IF (TotalLinesServedInGrid = 0, 0, CHOOSE (GridPlantIndex, 0, (1 + DS3DS1UnitsperTerminalLocation) * 4, (NumberDLCLsatFullCapacity + 1 + DS3DS1UnitsperTerminalLocation) * 4))	
BY	DLCSFibersThisGrid	# DLC-S Fibers this Grid		= CHOOSE (GridPlantIndex, 0, NumberFibersPerGrid, 0)	
BZ	DLCLFibersThisGrid	# DLC-L Fibers this Grid		= CHOOSE (GridPlantIndex, 0, 0, NumberFibersPerGrid)	
CA	AverageDLCtoFDIDistance	Average DLC to FDI Distance		= AVERAGE (ULQHorizontalfttoRoadCentroid + ULQverticalfttoRoadCentroid, LLQHorizontalfttoRoadCentroid + LLQverticalfttoRoadCentroid, URQHorizontalfttoRoadCentroid + URQverticalfttoRoadCentroid, LRQHorizontalfttoRoadCentroid + LRQverticalfttoRoadCentroid)	

CB	AverageFeederDistance	Average Feeder Distance		= MainFeederLength + SubfeederLength + SubfeederPart2Distance + AverageDLCtoFD1Distance	
CC	LargestQuadArea Number	Largest Quad Area Number		= IF (MAX (ULQArea, LLQArea, URQArea, LRQArea) = ULQArea, 1, IF (MAX (ULQArea, LLQArea, URQArea, LRQArea) = LLQArea, 2, IF (MAX (ULQArea, LLQArea, URQArea, LRQArea) = LRQArea, 3, 4)))	

Sheet: DLC to FDI

Col	Range Name	Column Name	Column Comment	Formula	Formula Comment
A	LeftSideCommoncableDistance	Left Side Common Cable Distance		= IF (ULQHorizontalfttoRoadCentroid < = LLQHorizontalfttoRoadCentroid, ULQHorizontalfttoRoadCentroid, LLQHorizontalfttoRoadCentroid) * SoilSlopeFactor	
B	ULQConnectingCableDistance	ULQ Connecting Cable Distance		= IF (ULQPairsrequired = 0, 0, (ULQHorizontalfttoRoadCentroid + ULQverticalfttoRoadCentroid) * SoilSlopeFactor - LeftSideCommoncableDistance)	
C	ULQWorkingpairs	ULQ Working Pairs		= ResidenceLineBalancingFctr * (ULQHouseholds * (VLOOKUP (StateId, StateInformationTable, 2) - 1) + ULQHousingUnits) + BusinessLineBalancingFctr * ULQBusinessLines	
D	ULQPairsrequired	ULQ Pairs Required		= ULQWorkingpairs / FeederFill	
E	ULQMaxSizeConnectingCables	ULQ Max Size Connecting Cables		= IF (ULQPairsrequired > MaxFeederSize, TRUNC (ULQPairsrequired / MaxFeederSize) , 0)	
F	ULQResidualCableSize	ULQ Residual Cable Size		= IF (ULQPairsrequired = 0, 0, INDEX (FeederCableSize, MATCH (ULQPairsrequired - MaxFeederSize * ULQMaxSizeConnectingCables, FeederCableSize, - 1) , 1))	
G	ULQUGCableDistance	ULQ Underground Cable Distance		= VLOOKUP (Density, CHOOSE (CopperDepthCondition, DistrHardMixTable, DistrSoftMixTable, DistrNormMixTable) , 2) * ULQConnectingCableDistance	
H	ULQDucts	ULQ Ducts		= IF (ULQUGCableDistance < 1, 0, ULQMaxSizeConnectingCables + 2)	
I	ULQManholes	ULQ Manholes		= IF (ULQUGCableDistance < 1, 0, ULQUGCableDistance / VLOOKUP (Density, DistrSpacingTable, 2))	
J	ULQStandardManholeCost	ULQ Standard Manhole Cost		= IF (OR (ULQManholes = 0, ULQDucts > 9) , 0, VLOOKUP (ULQDucts, CHOOSE (FiberDepthCondition, ManholeHardRockTable, ManholeSoftRockTable, ManholeNormalSurfaceTable) , DensityColumnPointer))	
K	ULQExpandedManholeCost	ULQ Expanded Manhole Cost		= IF (OR (ULQManholes = 0, ULQDucts < = 9) , 0, VLOOKUP (9, CHOOSE (FiberDepthCondition, ManholeHardRockTable, ManholeSoftRockTable, ManholeNormalSurfaceTable) , DensityColumnPointer) + TRUNC (ULQDucts / 9) * VLOOKUP (99, CHOOSE (FiberDepthCondition, ManholeHardRockTable, ManholeSoftRockTable, ManholeNormalSurfaceTable) , DensityColumnPointer))	
L	ULQConduitCost	ULQ Conduit Cost		= IF (ULQUGCableDistance < 1, 0, WaterCostMultiplier * ULQUGCableDistance * (IF (FiberDepthCondition = 1, VLOOKUP (Density, HardRockStructure, 2) , IF (OR (FiberDepthCondition = 2, SurfaceIndicator = 1) , VLOOKUP (Density, SoftRockStructure, 2) , VLOOKUP (Density, NormalStructure, 2))) + ULQDucts * VLOOKUP (0, ConduitDensityTable, DensityColumnPointer)) + ULQManholes * IF (ULQDucts > 9, ULQExpandedManholeCost, ULQStandardManholeCost))	
M	ULQAerialcableDistance	ULQ Aerial Cable Distance		= VLOOKUP (Density, CHOOSE (CopperDepthCondition, DistrHardMixTable, DistrSoftMixTable, DistrNormMixTable) , 4) * ULQConnectingCableDistance	
N	ULQCableGauge	ULQ Cable Gauge		= IF ((LeftSideCommoncableDistance + ULQConnectingCableDistance + SQRT (ULQArea) * 5280) > 10800, 24, 26)	
O	ULQAerialcableMaterial	ULQ Aerial Cable Material $		= IF (ULQResidualCableSize = 0, 0, ULQAerialcableDistance * WaterCostMultiplier * CopperCostRatio * (VLOOKUP (ULQResidualCableSize, IF (ULQCableGauge = 26, AERCopper26Table, AERCopper24Table) , DensityColumnPointer, FALSE) + ULQMaxSizeConnectingCables * VLOOKUP (MaxFeederSize, IF (ULQCableGauge = 26, AERCopper26Table, AERCopper24Table) , DensityColumnPointer, FALSE)))	
P	ULQBuriedCable Distance	ULQ Buried Cable Distance		= VLOOKUP (Density, CHOOSE (CopperDepthCondition, DistrHardMixTable, DistrSoftMixTable, DistrNormMixTable) , 3) * ULQConnectingCableDistance	
Q	ULQBuriedCable Material	ULQ Buried Cable Material $		= IF (ULQResidualCableSize = 0, 0, ULQBuriedCableDistance * WaterCostMultiplier * CopperCostRatio * (VLOOKUP (ULQResidualCableSize, IF (ULQCableGauge = 26, BURCopper26Table, BURCopper24Table) , DensityColumnPointer, FALSE) + ULQMaxSizeConnectingCables * VLOOKUP (MaxFeederSize, IF (ULQCableGauge = 26, BURCopper26Table, BURCopper24Table) , DensityColumnPointer, FALSE)))	
R	ULQUGCableMaterial	ULQ Underground Cable Material $		= IF (ULQResidualCableSize = 0, 0, ULQUGCableDistance * WaterCostMultiplier * CopperCostRatio * (VLOOKUP (ULQResidualCableSize, IF (ULQCableGauge = 26, UGCopper26Table, UGCopper24Table) , DensityColumnPointer, FALSE) + ULQMaxSizeConnectingCables * VLOOKUP (MaxFeederSize, IF (ULQCableGauge = 26, UGCopper26Table, UGCopper24Table) , DensityColumnPointer, FALSE)))	
S	LLQConnectingCableDistance	LLQ Connecting Cable Distance		= IF (LLQPairsRequired = 0, 0, (LLQHorizontalfttoRoadCentroid + LLQverticalfttoRoadCentroid) * SoilSlopeFactor - LeftSideCommoncableDistance)	
T	LLQWorkingPairs	LLQ Working Pairs		= ResidenceLineBalancingFctr * (LLQHouseholds * (VLOOKUP (StateId, StateInformationTable, 2) - 1) + LLQHousingUnits) + BusinessLineBalancingFctr * LLQBusinessLines	

U	LLQPairsRequired	LLQ Pairs Required		= LLQWorkingPairs / FeederFill	
V	LLQMaxSizeConnectingCables	LLQ Max Size Connecting Cables		= IF (LLQPairsRequired > MaxFeederSize, TRUNC (LLQPairsRequired / MaxFeederSize) , 0)	
W	LLQResidualCable Size	LLQ Residual Cable Size		= IF (LLQPairsRequired = 0, 0, INDEX (FeederCableSize, MATCH (LLQPairsRequired - LLQMaxSizeConnectingCables * MaxFeederSize, FeederCableSize, - 1) , 1))	
X	LLQUGCableDistance	LLQ Underground Cable Distance		= VLOOKUP (Density, CHOOSE (CopperDepthCondition, DistrHardMixTable, DistrSoftMixTable, DistrNormMixTable) , 2) * LLQConnectingCableDistance	
Y	LLQDucts	LLQ Ducts		= IF (LLQUGCableDistance = 0, 0, LLQMaxSizeConnectingCables + 2)	
Z	LLQManholes	LLQ Manholes		= IF (LLQUGCableDistance < 1, 0, LLQUGCableDistance / VLOOKUP (Density, DistrSpacingTable, 2))	
AA	LLQStandardManholeCost	LLQ Standard Manhole Cost		= IF (OR (LLQManholes = 0, LLQDucts > 9) , 0, VLOOKUP (LLQDucts, CHOOSE (FiberDepthCondition, ManholeHardRockTable, ManholeSoftRockTable, ManholeNormalSurfaceTable) , DensityColumnPointer))	
AB	LLQExpandedManholeCost	LLQ Expanded Manhole Cost		= IF (OR (LLQManholes = 0, LLQDucts < = 9) , 0, VLOOKUP (9, CHOOSE (FiberDepthCondition, ManholeHardRockTable, ManholeSoftRockTable, ManholeNormalSurfaceTable) , DensityColumnPointer) + TRUNC (LLQDucts / 9) * VLOOKUP (99, CHOOSE (FiberDepthCondition, ManholeHardRockTable, ManholeSoftRockTable, ManholeNormalSurfaceTable) , DensityColumnPointer))	
AC	LLQConduitCost	LLQ Conduit Cost		= IF (LLQUGCableDistance < 1, 0, WaterCostMultiplier * LLQUGCableDistance * (IF (FiberDepthCondition = 1, VLOOKUP (Density, HardRockStructure, 2) , IF (OR (FiberDepthCondition = 2, SurfaceIndicator = 1) , VLOOKUP (Density, SoftRockStructure, 2) , VLOOKUP (Density, NormalStructure, 2))) + LLQDucts * VLOOKUP (0, ConduitDensityTable, DensityColumnPointer)) + LLQManholes * IF (LLQDucts > 9, LLQExpandedManholeCost, LLQStandardManholeCost))	
AD	LLQAerialcableDistance	LLQ Aerial Cable Distance		= VLOOKUP (Density, CHOOSE (CopperDepthCondition, DistrHardMixTable, DistrSoftMixTable, DistrNormMixTable) , 4) * LLQConnectingCableDistance	
AE	LLQCableGauge	LLQ Cable Gauge		= IF ((LeftSideCommoncableDistance + LLQConnectingCableDistance + SQRT (LLQArea) * 5280) > 10800, 24, 26)	
AF	LLQAerialcableMaterial	LLQ Aerial Cable Material $		= IF (LLQResidualCableSize = 0, 0, LLQAerialCableDistance * WaterCostMultiplier * CopperCostRatio * (VLOOKUP (LLQResidualCableSize, IF (LLQCableGauge = 26, AERCopper26Table, AERCopper24Table) , DensityColumnPointer, FALSE) + LLQMaxSizeConnectingCables * VLOOKUP (MaxFeederSize, IF (LLQCableGauge = 26, AERCopper26Table, AERCopper24Table) , DensityColumnPointer, FALSE)))	
AG	LLQBuriedCableDistance	LLQ Buried Cable Distance		= VLOOKUP (Density, CHOOSE (CopperDepthCondition, DistrHardMixTable, DistrSoftMixTable, DistrNormMixTable) , 3) * LLQConnectingCableDistance	
AH	LLQBuriedCable Material	LLQ Buried Cable Material $		= IF (LLQResidualCableSize = 0, 0, LLQBuriedCableDistance * WaterCostMultiplier * CopperCostRatio * (VLOOKUP (LLQResidualCableSize, IF (LLQCableGauge = 26, BURCopper26Table, BURCopper24Table) , DensityColumnPointer, FALSE) + LLQMaxSizeConnectingCables * VLOOKUP (MaxFeederSize, IF (LLQCableGauge = 26, BURCopper26Table, BURCopper24Table) , DensityColumnPointer, FALSE)))	
AI	LLQUGCableMaterial	LLQ Underground Cable Material $		= IF (LLQResidualCableSize = 0, 0, LLQUGCableDistance * WaterCostMultiplier * CopperCostRatio * (VLOOKUP (LLQResidualCableSize, IF (LLQCableGauge = 26, UGCopper26Table, UGCopper24Table) , DensityColumnPointer, FALSE) + LLQMaxSizeConnectingCables * VLOOKUP (MaxFeederSize, IF (LLQCableGauge = 26, UGCopper26Table, UGCopper24Table) , DensityColumnPointer, FALSE)))	
AJ	LeftSideCommonCablePairsRequired	Left Side Common Cable Pairs Required		= ULQPairsrequired + LLQPairsRequired	
AK	NumberLeftCommonMaxSizeCables	Number Of Left Common Max Size Cables		= IF (LeftSideCommonCablePairsRequired > MaxFeederSize, CEILING (LeftSideCommonCablePairsRequired / MaxFeederSize, 1) , 0)	
AL	LeftSideCommonCableResidualSize	Left Side Common Cable Residual Size		= INDEX (FeederCableSize, MATCH (LeftSideCommonCablePairsRequired - (MaxFeederSize * NumberLeftCommonMaxSizeCables) , FeederCableSize, - 1) , 1)	
AM	LeftSideCommonUGCableDistance	Left Side Common Underground Cable Distance		= IF (LeftSideCommonCablePairsRequired = 0, 0, VLOOKUP (Density, CHOOSE (CopperDepthCondition, DistrHardMixTable, DistrSoftMixTable, DistrNormMixTable) , 2) * LeftSideCommoncableDistance)	
AN	LeftSideCommonDucts	Left Side Common Ducts		= IF (LeftSideCommonUGCableDistance = 0, 0, NumberLeftCommonMaxSizeCables + 2)	

AO	LeftSideCommon Manholes	Left Side Common Manholes		= IF (LeftSideCommonUGCableDistance < 1, 0, LeftSideCommonUGCableDistance / VLOOKUP (Density, DistrSpacingTable, 2))	
AP	LeftSideCommonStandardManholeCost	Left Side Common Standard Manhole Cost		= IF (OR (LeftSideCommonManholes = 0, LeftSideCommonDucts > 9) , 0, VLOOKUP (LeftSideCommonDucts, CHOOSE (FiberDepthCondition, ManholeHardRockTable, ManholeSoftRockTable, ManholeNormalSurfaceTable) , DensityColumnPointer))	
AQ	LeftSideCommonExpandedManholeCost	Left Side Common Expanded Manhole Cost		= IF (OR (LeftSideCommonManholes = 0, LeftSideCommonDucts < = 9) , 0, VLOOKUP (9, CHOOSE (FiberDepthCondition, ManholeHardRockTable, ManholeSoftRockTable, ManholeNormalSurfaceTable) , DensityColumnPointer) + TRUNC (LeftSideCommonDucts / 9) * VLOOKUP (99, CHOOSE (FiberDepthCondition, ManholeHardRockTable, ManholeSoftRockTable, ManholeNormalSurfaceTable) , DensityColumnPointer))	
AR	LeftSideCommonConduitCost	Left Side Common Conduit Cost		= IF (LeftSideCommonUGCableDistance < 1, 0, WaterCostMultiplier * LeftSideCommonUGCableDistance * (IF (FiberDepthCondition = 1, VLOOKUP (Density, HardRockStructure, 2) , IF (OR (FiberDepthCondition = 2, SurfaceIndicator = 1) , VLOOKUP (Density, SoftRockStructure, 2) , VLOOKUP (Density, NormalStructure, 2))) + LeftSideCommonDucts * VLOOKUP (0, ConduitDensityTable, DensityColumnPointer)) + LeftSideCommonManholes * IF (LeftSideCommonDucts > 9, LeftSideCommonExpandedManholeCost, LeftSideCommonStandardManholeCost))	
AS	LeftSideCommonAerialcableDistance	Left Side Common Aerial Cable Distance		= IF (LeftSideCommonCablePairsRequired = 0, 0, VLOOKUP (Density, CHOOSE (CopperDepthCondition, DistrHardMixTable, DistrSoftMixTable, DistrNormMixTable) , 4) * LeftSideCommoncableDistance)	
AT	LeftSideCommonCableGauge	Left Side Common Cable Gauge		= MIN (LLQCableGauge, ULQCableGauge)	
AU	LeftSideCommonAerialcableMaterial	Left Side Common Aerial Cable Material $		= IF (LeftSideCommonCableResidualSize = 0, 0, LeftSideCommonAerialcableDistance * WaterCostMultiplier * CopperCostRatio * (VLOOKUP (LeftSideCommonCableResidualSize, IF (LeftSideCommonCableGauge = 26, AERCopper26Table, AERCopper24Table) , DensityColumnPointer, FALSE) + NumberLeftCommonMaxSizeCables * VLOOKUP (MaxFeederSize, IF (LeftSideCommonCableGauge = 26, AERCopper26Table, AERCopper24Table) , DensityColumnPointer, FALSE)))	
AV	LeftSideCommonBuriedCableDistance	Left Side Common Buried Cable Distance		= IF (LeftSideCommonCablePairsRequired = 0, 0, VLOOKUP (Density, CHOOSE (CopperDepthCondition, DistrHardMixTable, DistrSoftMixTable, DistrNormMixTable) , 3) * LeftSideCommoncableDistance)	
AW	LeftSideCommonBuriedCableMaterial	Left Side Common Buried Cable Material $		= IF (LeftSideCommonCableResidualSize = 0, 0, LeftSideCommonBuriedCableDistance * WaterCostMultiplier * CopperCostRatio * (VLOOKUP (LeftSideCommonCableResidualSize, IF (LeftSideCommonCableGauge = 26, BURCopper26Table, BURCopper24Table) , DensityColumnPointer, FALSE) + NumberLeftCommonMaxSizeCables * VLOOKUP (MaxFeederSize, IF (LeftSideCommonCableGauge = 26, BURCopper26Table, BURCopper24Table) , DensityColumnPointer, FALSE)))	
AX	LeftSideCommonUGCableMaterial	Left Side Common Underground Cable Material $		= IF (LeftSideCommonCableResidualSize = 0, 0, LeftSideCommoncableDistance * WaterCostMultiplier * CopperCostRatio * (VLOOKUP (LeftSideCommonCableResidualSize, IF (LeftSideCommonCableGauge = 26, UGCopper26Table, UGCopper24Table) , DensityColumnPointer, FALSE) + NumberLeftCommonMaxSizeCables * VLOOKUP (MaxFeederSize, IF (LeftSideCommonCableGauge = 26, UGCopper26Table, UGCopper24Table) , DensityColumnPointer, FALSE)))	
AY	RightSideCommoncableDistance	Right Side Common Cable Distance		= IF (URQHorizontalfttoRoadCentroid < = LRQHorizontalfttoRoadCentroid, URQHorizontalfttoRoadCentroid, LRQHorizontalfttoRoadCentroid) * SoilSlopeFactor	
AZ	URQConnectingCableDistance	URQ Connecting Cable Distance		= IF (URQPairsrequired = 0, 0, (URQHorizontalfttoRoadCentroid + URQverticalfttoRoadCentroid) * SoilSlopeFactor - RightSideCommoncableDistance)	
BA	URQWorkingpairs	URQ Working Pairs		= ResidenceLineBalancingFctr * (URQHouseholds * (VLOOKUP (StateId, StateInformationTable, 2) - 1) + URQHousingUnits) + BusinessLineBalancingFctr * URQBusinessLines	
BB	URQPairsrequired	URQ Pairs Required		= URQWorkingpairs / FeederFill	
BC	URQMaxSizeConnectingCables	URQ Max Size Connecting Cables		= IF (URQPairsrequired > MaxFeederSize, TRUNC (URQPairsrequired / MaxFeederSize) , 0)	
BD	URQResidualCableSize	URQ Residual Cable Size		= IF (URQPairsrequired = 0, 0, INDEX (FeederCableSize, MATCH (URQPairsrequired - URQMaxSizeConnectingCables * MaxFeederSize, FeederCableSize, - 1) , 1))	
BE	URQUGCableDistance	URQ Underground Cable Distance		= VLOOKUP (Density, CHOOSE (CopperDepthCondition, DistrHardMixTable, DistrSoftMixTable, DistrNormMixTable) , 2) * URQConnectingCableDistance	

BF	URQDucts	URQ Ducts		= IF (URQUGCableDistance < 1, 0, URQMaxSizeConnectingCables + 2)	
BG	URQManholes	URQ Manholes		= IF (URQUGCableDistance < 1, 0, URQUGCableDistance / VLOOKUP (Density, DistrSpacingTable, 2))	
BH	URQStandardManholeCost	URQ Standard Manhole Cost		= IF (OR (URQManholes = 0, URQDucts > 9) , 0, VLOOKUP (URQDucts, CHOOSE (FiberDepthCondition, ManholeHardRockTable, ManholeSoftRockTable, ManholeNormalSurfaceTable) , DensityColumnPointer))	
BI	URQExpandedManholeCost	URQ Expanded Manhole Cost		= IF (OR (URQManholes = 0, URQDucts < = 9) , 0, VLOOKUP (9, CHOOSE (FiberDepthCondition, ManholeHardRockTable, ManholeSoftRockTable, ManholeNormalSurfaceTable) , DensityColumnPointer) + TRUNC (URQDucts / 9) * VLOOKUP (99, CHOOSE (FiberDepthCondition, ManholeHardRockTable, ManholeSoftRockTable, ManholeNormalSurfaceTable) , DensityColumnPointer))	
BJ	URQConduitCost	URQ Conduit Cost		= IF (URQUGCableDistance < 1, 0, WaterCostMultiplier * URQUGCableDistance * (IF (FiberDepthCondition = 1, VLOOKUP (Density, HardRockStructure, 2) , IF (OR (FiberDepthCondition = 2, SurfaceIndicator = 1) , VLOOKUP (Density, SoftRockStructure, 2) , VLOOKUP (Density, NormalStructure, 2))) + URQDucts * VLOOKUP (0, ConduitDensityTable, DensityColumnPointer)) + URQManholes * IF (URQDucts > 9, URQExpandedManholeCost, URQStandardManholeCost))	
BK	URQAerialcableDistance	URQ Aerial Cable Distance		= VLOOKUP (Density, CHOOSE (CopperDepthCondition, DistrHardMixTable, DistrSoftMixTable, DistrNormMixTable) , 4) * URQConnectingCableDistance	
BL	URQCableGauge	URQ Cable Gauge		= IF ((RightSideCommoncableDistance + URQConnectingCableDistance + SQRT (URQArea) * 5280) > 10800, 24, 26)	
BM	URQAerialcableMaterial	URQ Aerial Cable Material $		= IF (URQResidualCableSize = 0, 0, URQAerialcableDistance * WaterCostMultiplier * CopperCostRatio * (VLOOKUP (URQResidualCableSize, IF (URQCableGauge = 26, AERCopper26Table, AERCopper24Table) , DensityColumnPointer, FALSE) + URQMaxSizeConnectingCables * VLOOKUP (MaxFeederSize, IF (URQCableGauge = 26, AERCopper26Table, AERCopper24Table) , DensityColumnPointer, FALSE)))	
BN	URQBuriedCable Distance	URQ Buried Cable Distance		= VLOOKUP (Density, CHOOSE (CopperDepthCondition, DistrHardMixTable, DistrSoftMixTable, DistrNormMixTable) , 3) * URQConnectingCableDistance	
BO	URQBuriedCable Material	URQ Buried Cable Material $		= IF (URQResidualCableSize = 0, 0, URQBuriedCableDistance * WaterCostMultiplier * CopperCostRatio * (VLOOKUP (URQResidualCableSize, IF (URQCableGauge = 26, BURCopper26Table, BURCopper24Table) , DensityColumnPointer, FALSE) + URQMaxSizeConnectingCables * VLOOKUP (MaxFeederSize, IF (URQCableGauge = 26, BURCopper26Table, BURCopper24Table) , DensityColumnPointer, FALSE)))	
BP	URQUGCableMaterial	URQ Underground Cable Material $		= IF (URQResidualCableSize = 0, 0, URQUGCableDistance * WaterCostMultiplier * CopperCostRatio * (VLOOKUP (URQResidualCableSize, IF (URQCableGauge = 26, UGCopper26Table, UGCopper24Table) , DensityColumnPointer, FALSE) + URQMaxSizeConnectingCables * VLOOKUP (MaxFeederSize, IF (URQCableGauge = 26, UGCopper26Table, UGCopper24Table) , DensityColumnPointer, FALSE)))	
BQ	LRQConnectingCableDistance	LRQ Connecting Cable Distance		= IF (LRQPairsRequired = 0, 0, (LRQHorizontalftoRoadCentroid + LRQverticalftoRoadCentroid) * SoilSlopeFactor - RightSideCommoncableDistance)	
BR	LRQWorkingPairs	LRQ Working Pairs		= ResidenceLineBalancingFctr * (LRQHouseholds * (VLOOKUP (StateId, StateInformationTable, 2) - 1) + LRQHousingUnits) + BusinessLineBalancingFctr * LRQBusinessLines	
BS	LRQPairsRequired	LRQ Pairs Required		= LRQWorkingPairs / FeederFill	
BT	LRQMaxSizeConnectingCables	LRQ Max Size Connecting Cables		= IF (LRQPairsRequired > MaxFeederSize, TRUNC (LRQPairsRequired / MaxFeederSize) , 0)	
BU	LRQResidualCableSize	LRQ Residual Cable Size		= IF (LRQPairsRequired = 0, 0, INDEX (FeederCableSize, MATCH (LRQPairsRequired - LRQMaxSizeConnectingCables * MaxFeederSize, FeederCableSize, - 1) , 1))	
BV	LRQUGCableDistance	LRQ Underground Cable Distance		= VLOOKUP (Density, CHOOSE (CopperDepthCondition, DistrHardMixTable, DistrSoftMixTable, DistrNormMixTable) , 2) * LRQConnectingCableDistance	
BW	LRQDucts	LRQ Ducts		= IF (LRQUGCableDistance = 0, 0, LRQMaxSizeConnectingCables + 2)	
BX	LRQManholes	LRQ Manholes		= IF (LRQUGCableDistance < 1, 0, LRQUGCableDistance / VLOOKUP (Density, DistrSpacingTable, 2))	
BY	LRQStandardManholeCost	LRQ Standard Manhole Cost		= IF (OR (LRQManholes = 0, LRQDucts > 9) , 0, VLOOKUP (LRQDucts, CHOOSE (FiberDepthCondition, ManholeHardRockTable, ManholeSoftRockTable, ManholeNormalSurfaceTable) , DensityColumnPointer))	
BZ	LRQExpandedManholeCost	LRQ Expanded Manhole Cost		= IF (OR (LRQManholes = 0, LRQDucts < = 9) , 0, VLOOKUP (9, CHOOSE (FiberDepthCondition, ManholeHardRockTable, ManholeSoftRockTable, ManholeNormalSurfaceTable) , DensityColumnPointer) + TRUNC (LRQDucts / 9) * VLOOKUP (99, CHOOSE (FiberDepthCondition, ManholeHardRockTable, ManholeSoftRockTable, ManholeNormalSurfaceTable) , DensityColumnPointer))	

CA	LRQConduitCost	LRQ Conduit Cost		= IF (LRQUGCableDistance < 1, 0, WaterCostMultiplier * LRQUGCableDistance * (IF (FiberDepthCondition = 1, VLOOKUP (Density, HardRockStructure, 2) , IF (OR (FiberDepthCondition = 2, SurfaceIndicator = 1) , VLOOKUP (Density, SoftRockStructure, 2) , VLOOKUP (Density, NormalStructure, 2))) + LRQDucts * VLOOKUP (0, ConduitDensityTable, DensityColumnPointer)) + LRQManholes * IF (LRQDucts > 9, LRQExpandedManholeCost, LRQStandardManholeCost))	
CB	LRQAerialcableDistance	LRQ Aerial Cable Distance		= VLOOKUP (Density, CHOOSE (CopperDepthCondition, CopperHardMixTable, DistrHardMixTable, DistrSoftMixTable, DistrNormMixTable) , 4) * LRQConnectingCableDistance	
CC	LRQCableGauge	LRQ Cable Gauge		= IF ((RightSideCommoncableDistance + LRQConnectingCableDistance + SQRT (LRQArea) * 5280) > 10800, 24, 26)	
CD	LRQAerialcableMaterial	LRQ Aerial Cable Material $		= IF (LRQResidualCableSize = 0, 0, LRQAerialCableDistance * WaterCostMultiplier * CopperCostRatio * (VLOOKUP (LRQResidualCableSize, IF (LRQCableGauge = 26, AERCopper26Table, AERCopper24Table) , DensityColumnPointer, FALSE) + LRQMaxSizeConnectingCables * VLOOKUP (MaxFeederSize, IF (LRQCableGauge = 26, AERCopper26Table, AERCopper24Table) , DensityColumnPointer, FALSE)))	
CE	LRQBuriedCableDistance	LRQ Buried Cable Distance		= VLOOKUP (Density, CHOOSE (CopperDepthCondition, DistrHardMixTable, DistrSoftMixTable, DistrNormMixTable) , 3) * LRQConnectingCableDistance	
CF	LRQBuriedCable Material	LRQ Buried Cable Material $		= IF (LRQResidualCableSize = 0, 0, LRQBuriedCableDistance * WaterCostMultiplier * CopperCostRatio * (VLOOKUP (LRQResidualCableSize, IF (LRQCableGauge = 26, BURCopper26Table, BURCopper24Table) , DensityColumnPointer, FALSE) + LRQMaxSizeConnectingCables * VLOOKUP (MaxFeederSize, IF (LRQCableGauge = 26, BURCopper26Table, BURCopper24Table) , DensityColumnPointer, FALSE)))	
CG	LRQUGCableMaterial	LRQ Underground Cable Material $		= IF (LRQResidualCableSize = 0, 0, LRQUGCableDistance * WaterCostMultiplier * CopperCostRatio * (VLOOKUP (LRQResidualCableSize, IF (LRQCableGauge = 26, UGCopper26Table, UGCopper24Table) , DensityColumnPointer, FALSE) + LRQMaxSizeConnectingCables * VLOOKUP (MaxFeederSize, IF (LRQCableGauge = 26, UGCopper26Table, UGCopper24Table) , DensityColumnPointer, FALSE)))	
CH	RightSideCommonCablePairsRequired	Right Side Common Cable Pairs Required		= LRQPairsRequired + URQPairsrequired	
CI	NumberMaxSizeRightCommonCables	Number Of Max Size Right Common Cables		= IF (RightSideCommonCablePairsRequired > MaxFeederSize, CEILING (RightSideCommonCablePairsRequired / MaxFeederSize, 1) , 0)	
CJ	RightSideCommonCableResidualSize	Right Side Common Cable Residual Size		= INDEX (FeederCableSize, MATCH (RightSideCommonCablePairsRequired - NumberMaxSizeRightCommonCables * MaxFeederSize, FeederCableSize, - 1) , 1)	
CK	RightSideCommonUGCableDistance	Right Side Common Underground Cable Distance		= IF (RightSideCommonCablePairsRequired = 0, 0, VLOOKUP (Density, CHOOSE (CopperDepthCondition, DistrHardMixTable, DistrSoftMixTable, DistrNormMixTable) , 2) * RightSideCommoncableDistance)	
CL	RightSideCommonDucts	Right Side Common Ducts		= IF (RightSideCommonUGCableDistance = 0, 0, NumberMaxSizeRightCommonCables + 2)	
CM	RightSideCommonManholes	Right Side Common Manholes		= IF (RightSideCommonUGCableDistance < 1, 0, RightSideCommonUGCableDistance / VLOOKUP (Density, DistrSpacingTable, 2))	
CN	RightSideCommonStandardManholeCost	Right Side Common Standard Manhole Cost		= IF (OR (RightSideCommonManholes = 0, RightSideCommonDucts > 9) , 0, VLOOKUP (RightSideCommonDucts, CHOOSE (FiberDepthCondition, ManholeHardRockTable, ManholeSoftRockTable, ManholeNormalSurfaceTable) , DensityColumnPointer))	
CO	RightSideCommonExpandedManholeCost	Right Side Common Expanded Manhole Cost		= IF (OR (RightSideCommonManholes = 0, RightSideCommonDucts < = 9) , 0, VLOOKUP (9, CHOOSE (FiberDepthCondition, ManholeHardRockTable, ManholeSoftRockTable, ManholeNormalSurfaceTable) , DensityColumnPointer) + TRUNC (RightSideCommonDucts / 9) * VLOOKUP (99, CHOOSE (FiberDepthCondition, ManholeHardRockTable, ManholeSoftRockTable, ManholeNormalSurfaceTable) , DensityColumnPointer))	
CP	RightSideCommonConduitCost	Right Side Common Conduit Cost		= IF (RightSideCommonUGCableDistance < 1, 0, WaterCostMultiplier * RightSideCommonUGCableDistance * (IF (FiberDepthCondition = 1, VLOOKUP (Density, HardRockStructure, 2) , IF (OR (FiberDepthCondition = 2, SurfaceIndicator = 1) , VLOOKUP (Density, SoftRockStructure, 2) , VLOOKUP (Density, NormalStructure, 2))) + RightSideCommonDucts * VLOOKUP (0, ConduitDensityTable, DensityColumnPointer)) + RightSideCommonManholes * IF (RightSideCommonDucts > 9, RightSideCommonExpandedManholeCost, RightSideCommonStandardManholeCost))	

CQ	RightSideCommonAerialcableDistance	Right Side Common Aerial Cable Distance		= IF (RightSideCommonCablePairsRequired = 0, 0, VLOOKUP (Density, CHOOSE (CopperDepthCondition, DistrHardMixTable, DistrSoftMixTable, DistrNormMixTable) , 4) * RightSideCommoncableDistance)	
CR	RightSideCommonCableGauge	Right Side Common Cable Gauge		= MIN (LRQCableGauge, URQCableGauge)	
CS	RightSideCommonAerialcableMaterial	Right Side Common Aerial Cable Material $		= IF (RightSideCommonCableResidualSize = 0, 0, RightSideCommonAerialcableDistance * WaterCostMultiplier * CopperCostRatio * (VLOOKUP (RightSideCommonCableResidualSize, IF (RightSideCommonCableGauge = 26, AERCopper26Table, AERCopper24Table) , DensityColumnPointer, FALSE) + NumberMaxSizeRightCommonCables * VLOOKUP (MaxFeederSize, IF (RightSideCommonCableGauge = 26, AERCopper26Table, AERCopper24Table) , DensityColumnPointer, FALSE)))	
CT	RightSideCommonBuriedCableDistance	Right Side Common Buried Cable Distance		= IF (RightSideCommonCablePairsRequired = 0, 0, VLOOKUP (Density, CHOOSE (CopperDepthCondition, DistrHardMixTable, DistrSoftMixTable, DistrNormMixTable) , 3) * RightSideCommoncableDistance)	
CU	RightSideCommonBuriedCableMaterial	Right Side Common Buried Cable Material $		= IF (RightSideCommonCableResidualSize = 0, 0, RightSideCommonBuriedCableDistance * WaterCostMultiplier * CopperCostRatio * (VLOOKUP (RightSideCommonCableResidualSize, IF (RightSideCommonCableGauge = 26, BURCopper26Table, BURCopper24Table) , DensityColumnPointer, FALSE) + NumberMaxSizeRightCommonCables * VLOOKUP (MaxFeederSize, IF (RightSideCommonCableGauge = 26, BURCopper26Table, BURCopper24Table) , DensityColumnPointer, FALSE)))	
CV	RightSideCommonUGCableMaterial	Right Side Common Underground Cable Material $		= IF (RightSideCommonCableResidualSize = 0, 0, RightSideCommonUGCableDistance * WaterCostMultiplier * CopperCostRatio * (VLOOKUP (RightSideCommonCableResidualSize, IF (RightSideCommonCableGauge = 26, UGCopper26Table, UGCopper24Table) , DensityColumnPointer, FALSE) + NumberMaxSizeRightCommonCables * VLOOKUP (MaxFeederSize, IF (RightSideCommonCableGauge = 26, UGCopper26Table, UGCopper24Table) , DensityColumnPointer, FALSE)))	
CW	TotalDLCFDIAerialCableMaterial	Total DLC/FDI Aerial Cable Material $		= ULQAerialcableMaterial + LLQAerialCableMaterial + LeftSideCommonAerialcableMaterial + URQAerialcableMaterial + LRQAerialCableMaterial + RightSideCommonAerialcableMaterial	
CX	TotalDLCFDIBuriedCableMaterial	Total DLC/FDI Buried Cable Material $		= ULQBuriedCableMaterial + LLQBuriedCableMaterial + LeftSideCommonBuriedCableMaterial + URQBuriedCableMaterial + LRQBuriedCableMaterial + RightSideCommonBuriedCableMaterial	
CY	TotalDLCFDIUGCableMaterial	Total DLC/FDI Underground Cable Material $		= ULQUGCableMaterial + LLQUGCableMaterial + LeftSideCommonUGCableMaterial + URQUGCableMaterial + LRQUGCableMaterial + RightSideCommonUGCableMaterial	
CZ	TotalDLCFDICablematerial	Total DLC/FDI Cable Material $		= TotalDLCFDIAerialCableMaterial + TotalDLCFDIBuriedCableMaterial + TotalDLCFDIUGCableMaterial	
DA	TotalAerialDLCFDICableDistance	Total Aerial DLC/FDI Cable Distance		= ULQAerialcableDistance + LLQAerialCableDistance + LeftSideCommoncableDistance + URQAerialcableDistance + LRQAerialCableDistance + RightSideCommonAerialcableDistance	
DB	NumberofDLCFDIPoles	Number Of DLC/FDI Poles		= IF (TotalAerialDLCFDICableDistance = 0, 0, TotalAerialDLCFDICableDistance / VLOOKUP (Density, DistrSpacingTable, 3))	
DC	TotalAerialDLCFDICableStructure	Total Aerial DLC/FDI Cable Structure $		= NumberofDLCFDIPoles * WaterCostMultiplier * VLOOKUP (Density, CHOOSE (FiberDepthCondition, HardRockStructure, SoftRockStructure, NormalStructure) , 6)	
DD	TotalburiedDLCFDICableDistance	Total buried DLC/FDI Cable Distance		= ULQBuriedCableDistance + LLQBuriedCableDistance + LeftSideCommonBuriedCableDistance + URQBuriedCableDistance + LRQBuriedCableDistance + RightSideCommonBuriedCableDistance	
DE	TotalBuriedDLCFDICableStructure	Total Buried DLC/FDI Cable Structure $		= IF (TotalburiedDLCFDICableDistance < 1, 0, WaterCostMultiplier * TotalburiedDLCFDICableDistance * VLOOKUP (Density, CHOOSE (CopperDepthCondition, HardRockStructure, SoftRockStructure, NormalStructure) , 4))	
DF	TotalUGDLCFDICableStructure	Total Underground DLC/FDI Cable Structure $		= ULQConduitCost + LLQConduitCost + LeftSideCommonConduitCost + URQConduitCost + LRQConduitCost + RightSideCommonConduitCost	

DG	TotalDLCFDICableStructure	Total DLC/FDI Cable Structure $		= TotalAerialDLCFDICableStructure + TotalBuriedDLCFDICableStructure + TotalUGDLCFDICableStructure	
DH	TotalDLCtoFDICableCost	Total DLC to FDI Cable Cost		= TotalDLCFDICablematerial + TotalDLCFDICableStructure	

Sheet: New Distribution

Col	Range Name	Column Name	Column Comment	Formula	Formula Comment
A	PairsperULQbranchcable	Pairs Per ULQ Branch Cable		= IF (NumberULQbranchcables = 0, 0, CEILING (NumberpairsinULquadrant / (2 * NumberULQbranchcables) / DistributionFillFactor, 1))	
B	ULQMaxSizeBranchCables	ULQ Max Size Branch Cables		= IF (PairsperULQbranchcable > MaxDistrSize, TRUNC (PairsperULQbranchcable / MaxDistrSize) , 0)	
C	ULQResidualBranchCableSize	ULQ Residual Branch Cable Size		= IF (PairsperULQbranchcable = 0, 0, INDEX (DistrCableSize, MATCH (PairsperULQbranchcable - ULQMaxSizeBranchCables * MaxDistrSize, DistrCableSize, - 1) , 1))	
D	ULQMaxSizeBackboneCables	ULQ Max Size Backbone Cables		= NumberULQbranchcables * ULQMaxSizeBranchCables + IF (NumberULQbranchcables * ULQResidualBranchCableSize > MaxDistrSize, TRUNC (NumberULQbranchcables * ULQResidualBranchCableSize / MaxDistrSize) , 0)	
E	PreULQResidualBackboneCableSize	ULQ Residual Backbone Cable Size		= IF (ULQResidualBranchCableSize = 0, 0, INDEX (DistrCableSize, MATCH (IF (NumberULQbranchcables * ULQResidualBranchCableSize > MaxDistrSize, NumberULQbranchcables * ULQResidualBranchCableSize - MaxDistrSize * TRUNC (NumberULQbranchcables * ULQResidualBranchCableSize / MaxDistrSize) , NumberULQbranchcables * ULQResidualBranchCableSize) , DistrCableSize, - 1) , 1))	
F	ULQBranchBackboneAerialDistance	ULQ Branch & Backbone Aerial Distance		= VLOOKUP (Density, CHOOSE (CopperDepthCondition, DistrHardMixTable, DistrSoftMixTable, DistrNormMixTable) , 4) * ULQBackboneandBranchCableLength	
G	ULQBranchAerialCableMaterial	ULQ Branch Aerial Cable Material $		= IF (ULQResidualBranchCableSize = 0, 0, NumberULQbranchcables * ULQBranchBackboneAerialDistance * WaterCostMultiplier * CopperCostRatio * (VLOOKUP (ULQResidualBranchCableSize, IF (ULQCableGauge = 26, AERCopper26Table, AERCopper24Table) , DensityColumnPointer, FALSE) + ULQMaxSizeBranchCables * VLOOKUP (MaxDistrSize, IF (ULQCableGauge = 26, AERCopper26Table, AERCopper24Table) , DensityColumnPointer, FALSE)))	
H	ULQBackboneAerialCableMaterial	ULQ Backbone Aerial Cable Material $		= IF (ULQResidualBackboneCableSize = 0, 0, ULQBranchBackboneAerialDistance * WaterCostMultiplier * CopperCostRatio * (VLOOKUP (ULQResidualBackboneCableSize, IF (ULQCableGauge = 26, AERCopper26Table, AERCopper24Table) , DensityColumnPointer, FALSE) + ULQMaxSizeBackboneCables * VLOOKUP (MaxFeederSize, IF (ULQCableGauge = 26, AERCopper26Table, AERCopper24Table) , DensityColumnPointer, FALSE)))	
I	ULQBranchBackboneBuriedDistance	ULQ Branch & Backbone Buried Distance		= VLOOKUP (Density, CHOOSE (CopperDepthCondition, DistrHardMixTable, DistrSoftMixTable, DistrNormMixTable) , 3) * ULQBackboneandBranchCableLength	
J	ULQBranchBuriedCableMaterial	ULQ Branch Buried Cable Material $		= IF (ULQResidualBranchCableSize = 0, 0, NumberULQbranchcables * ULQBranchBackboneBuriedDistance * WaterCostMultiplier * CopperCostRatio * (VLOOKUP (ULQResidualBranchCableSize, IF (ULQCableGauge = 26, BURCopper26Table, BURCopper24Table) , DensityColumnPointer, FALSE) + ULQMaxSizeBranchCables * VLOOKUP (MaxDistrSize, IF (ULQCableGauge = 26, BURCopper26Table, BURCopper24Table) , DensityColumnPointer, FALSE)))	
K	ULQBackboneBuriedCableMaterial	ULQ Backbone Buried Cable Material $		= IF (ULQResidualBackboneCableSize = 0, 0, ULQBranchBackboneBuriedDistance * WaterCostMultiplier * CopperCostRatio * (VLOOKUP (ULQResidualBackboneCableSize, IF (ULQCableGauge = 26, BURCopper26Table, BURCopper24Table) , DensityColumnPointer, FALSE) + ULQMaxSizeBackboneCables * VLOOKUP (MaxDistrSize, IF (ULQCableGauge = 26, BURCopper26Table, BURCopper24Table) , DensityColumnPointer, FALSE)))	
L	ULQBranchBackboneUGDistance	ULQ Branch & Backbone Underground Distance		= VLOOKUP (Density, CHOOSE (CopperDepthCondition, DistrHardMixTable, DistrSoftMixTable, DistrNormMixTable) , 2) * ULQBackboneandBranchCableLength	
M	ULQBranchUGCableMaterial	ULQ Branch Underground Cable Material $		= IF (ULQResidualBranchCableSize = 0, 0, NumberULQbranchcables * ULQBranchBackboneUGDistance * WaterCostMultiplier * CopperCostRatio * (VLOOKUP (ULQResidualBranchCableSize, IF (ULQCableGauge = 26, UGCopper26Table, UGCopper24Table) , DensityColumnPointer, FALSE) + ULQMaxSizeBranchCables * VLOOKUP (MaxDistrSize, IF (ULQCableGauge = 26, UGCopper26Table, UGCopper24Table) , DensityColumnPointer, FALSE)))	
N	ULQBackboneUGCableMaterial	ULQ Backbone Underground Cable Material $		= IF (ULQResidualBackboneCableSize = 0, 0, ULQBranchBackboneUGDistance * WaterCostMultiplier * CopperCostRatio * (VLOOKUP (ULQResidualBackboneCableSize, IF (ULQCableGauge = 26, UGCopper26Table, UGCopper24Table) , DensityColumnPointer, FALSE) + ULQMaxSizeBackboneCables * VLOOKUP (MaxDistrSize, IF (ULQCableGauge = 26, UGCopper26Table, UGCopper24Table) , DensityColumnPointer, FALSE)))	
O	ULQBranchDucts	Number ULQ Branch Ducts		= ULQMaxSizeBranchCables + 2	
P	ULQBranchManholes	Number ULQ Branch Manholes		= NumberULQbranchcables * IF (ULQBranchBackboneUGDistance < 1, 0, ULQBranchBackboneUGDistance / VLOOKUP (Density, DistrSpacingTable, 2))	

Q	ULQStandardBranchManholeCost	ULQ Standard Branch Manhole Cost		= IF (OR (ULQBranchManholes = 0, ULQBranchDucts > 9) , 0, VLOOKUP (ULQBranchDucts, CHOOSE (FiberDepthCondition, ManholeHardRockTable, ManholeSoftRockTable, ManholeNormalSurfaceTable) , DensityColumnPointer))	
R	ULQExpandedBranchManholeCost	ULQ Expanded Branch Manhole Cost		= IF (OR (ULQBranchManholes = 0, ULQBranchDucts < = 9) , 0, VLOOKUP (9, CHOOSE (FiberDepthCondition, ManholeHardRockTable, ManholeSoftRockTable, ManholeNormalSurfaceTable) , DensityColumnPointer) + TRUNC (ULQBranchDucts / 9) * VLOOKUP (99, CHOOSE (FiberDepthCondition, ManholeHardRockTable, ManholeSoftRockTable, ManholeNormalSurfaceTable) , DensityColumnPointer))	
S	ULQBackboneDucts	Number ULQ Backbone Ducts		= ULQMaxSizeBackboneCables + 2	
T	ULQBackboneManholes	Number ULQ Backbone Manholes		= IF (ULQBranchBackboneUGDistance < 1, 0, ULQBranchBackboneUGDistance / 2 / VLOOKUP (Density, DistrSpacingTable, 2))	
U	ULQStandardBackboneManholeCost	ULQ Standard Backbone Manhole Cost		= IF (OR (ULQBackboneManholes = 0, ULQBackboneDucts > 9) , 0, VLOOKUP (ULQBackboneDucts, CHOOSE (FiberDepthCondition, ManholeHardRockTable, ManholeSoftRockTable, ManholeNormalSurfaceTable) , DensityColumnPointer))	
V	ULQExpandedBackboneManholeCost	ULQ Expanded Backbone Manhole Cost		= IF (OR (ULQBackboneManholes = 0, ULQBackboneDucts < = 9) , 0, VLOOKUP (9, CHOOSE (FiberDepthCondition, ManholeHardRockTable, ManholeSoftRockTable, ManholeNormalSurfaceTable) , DensityColumnPointer) + TRUNC (ULQBackboneDucts / 9) * VLOOKUP (99, CHOOSE (FiberDepthCondition, ManholeHardRockTable, ManholeSoftRockTable, ManholeNormalSurfaceTable) , DensityColumnPointer))	
W	ULQUGStructure Cost	ULQ Underground Structure Cost		= ULQBranchManholes * IF (ULQBranchDucts < = 9, ULQStandardBranchManholeCost, ULQExpandedBranchManholeCost) + ULQBackboneManholes * IF (ULQBackboneDucts < = 9, ULQStandardBackboneManholeCost, ULQExpandedBackboneManholeCost) + (NumberULQbranchcables * ULQBranchDucts + ULQBackboneDucts) * ULQBranchBackboneUGDistance * VLOOKUP (0, ConduitDensityTable, DensityColumnPointer) + (NumberULQbranchcables + 1 / 2) * ULQBranchBackboneUGDistance * VLOOKUP (Density, CHOOSE (FiberDepthCondition, HardRockStructure, SoftRockStructure, NormalStructure) , 3)	
X	PairsperLLQbranchcable	Pairs Per LLQ Branch Cable		= IF (NumberLLQbranchcables = 0, 0, CEILING (NumberpairsinLLquadrant / (2 * NumberLLQbranchcables) / DistributionFillFactor, 1))	
Y	LLQMaxSizeBranchCables	LLQ Max Size Branch Cables		= IF (PairsperLLQbranchcable > MaxDistrSize, TRUNC (PairsperLLQbranchcable / MaxDistrSize) , 0)	
Z	LLQResidualBranchCableSize	LLQ Residual Branch Cable Size		= IF (PairsperLLQbranchcable = 0, 0, INDEX (DistrCableSize, MATCH (PairsperLLQbranchcable - LLQMaxSizeBranchCables * MaxDistrSize, DistrCableSize, - 1) , 1))	
AA	LLQMaxSizeBackboneCables	LLQ Max Size Backbone Cables		= NumberLLQbranchcables * LLQMaxSizeBranchCables + IF (NumberLLQbranchcables * LLQResidualBranchCableSize > MaxDistrSize, TRUNC (NumberLLQbranchcables * LLQResidualBranchCableSize / MaxDistrSize) , 0)	
AB	LLQResidualBackboneCableSize	LLQ Residual Backbone Cable Size		= IF (LLQResidualBranchCableSize = 0, 0, INDEX (DistrCableSize, MATCH (IF (NumberLLQbranchcables * LLQResidualBranchCableSize > MaxDistrSize, NumberLLQbranchcables * LLQResidualBranchCableSize - MaxDistrSize * TRUNC (NumberLLQbranchcables * LLQResidualBranchCableSize / MaxDistrSize) , NumberLLQbranchcables * LLQResidualBranchCableSize) , DistrCableSize, - 1) , 1))	
AC	LLQBranchBackboneAerialDistance	LLQ Branch & Backbone Aerial Distance		= VLOOKUP (Density, CHOOSE (CopperDepthCondition, DistrHardMixTable, DistrSoftMixTable, DistrNormMixTable) , 4) * LLQBackboneBranchCableLength	
AD	LLQBranchAerialCableMaterial	LLQ Branch Aerial Cable Material $		= IF (LLQResidualBranchCableSize = 0, 0, NumberLLQbranchcables * LLQBranchBackboneAerialDistance * WaterCostMultiplier * CopperCostRatio * (VLOOKUP (LLQResidualBranchCableSize, IF (LLQCableGauge = 26, AERCopper26Table, AERCopper24Table) , DensityColumnPointer, FALSE) + LLQMaxSizeBranchCables * VLOOKUP (MaxDistrSize, IF (LLQCableGauge = 26, AERCopper26Table, AERCopper24Table) , DensityColumnPointer, FALSE)))	
AE	LLQBackboneAerialCableMaterial	LLQ Backbone Aerial Cable Material $		= IF (LLQResidualBackboneCableSize = 0, 0, LLQBranchBackboneAerialDistance * WaterCostMultiplier * CopperCostRatio * (VLOOKUP (LLQResidualBackboneCableSize, IF (LLQCableGauge = 26, AERCopper26Table, AERCopper24Table) , DensityColumnPointer, FALSE) + LLQMaxSizeBackboneCables * VLOOKUP (MaxFeederSize, IF (LLQCableGauge = 26, AERCopper26Table, AERCopper24Table) , DensityColumnPointer, FALSE)))	
AF	LLQBranchBackboneBuriedDistance	LLQ Branch & Backbone Buried Distance		= VLOOKUP (Density, CHOOSE (CopperDepthCondition, DistrHardMixTable, DistrSoftMixTable, DistrNormMixTable) , 3) * LLQBackboneBranchCableLength	

AG	LLQBranchBuried CableMaterial	LLQ Branch Buried Cable Material $		= IF (LLQResidualBranchCableSize = 0, 0, NumberLLQbranchcables * LLQBranchBackboneBuriedDistance * WaterCostMultiplier * CopperCostRatio * (VLOOKUP (LLQResidualBranchCableSize, IF (LLQCableGauge = 26, BURCopper26Table, BURCopper24Table) , DensityColumnPointer, FALSE) + LLQMaxSizeBranchCables * VLOOKUP (MaxDistrSize, IF (LLQCableGauge = 26, BURCopper26Table, BURCopper24Table) , DensityColumnPointer, FALSE)))	
AH	LLQBackboneBuri edCableMaterial	LLQ Backbone Buried Cable Material $		= IF (LLQResidualBackboneCableSize = 0, 0, LLQBranchBackboneBuriedDistance * WaterCostMultiplier * CopperCostRatio * (VLOOKUP (LLQResidualBackboneCableSize, IF (LLQCableGauge = 26, BURCopper26Table, BURCopper24Table) , DensityColumnPointer, FALSE) + LLQMaxSizeBackboneCables * VLOOKUP (MaxDistrSize, IF (LLQCableGauge = 26, BURCopper26Table, BURCopper24Table) , DensityColumnPointer, FALSE)))	
AI	LLQBranchBackb oneUGDistance	LLQ Branch & Backbone Underground Distance		= VLOOKUP (Density, CHOOSE (CopperDepthCondition, DistrHardMixTable, DistrSoftMixTable, DistrNormMixTable) , 2) * LLQBackboneBranchCableLength	
AJ	LLQBranchUGCa bleMaterial	LLQ Branch Underground Cable Material $		= IF (LLQResidualBranchCableSize = 0, 0, NumberLLQbranchcables * LLQBranchBackboneUGDistance * WaterCostMultiplier * CopperCostRatio * (VLOOKUP (LLQResidualBranchCableSize, IF (LLQCableGauge = 26, UGCopper26Table, UGCopper24Table) , DensityColumnPointer, FALSE) + LLQMaxSizeBranchCables * VLOOKUP (MaxDistrSize, IF (LLQCableGauge = 26, UGCopper26Table, UGCopper24Table) , DensityColumnPointer, FALSE)))	
AK	LLQBackboneUG CableMaterial	LLQ Backbone Underground Cable Material $		= IF (LLQResidualBackboneCableSize = 0, 0, LLQBranchBackboneUGDistance * WaterCostMultiplier * CopperCostRatio * (VLOOKUP (LLQResidualBackboneCableSize, IF (LLQCableGauge = 26, UGCopper26Table, UGCopper24Table) , DensityColumnPointer, FALSE) + LLQMaxSizeBackboneCables * VLOOKUP (MaxDistrSize, IF (LLQCableGauge = 26, UGCopper26Table, UGCopper24Table) , DensityColumnPointer, FALSE)))	
AL	LLQBranchDucts	Number LLQ Branch Ducts		= LLQMaxSizeBranchCables + 2	
AM	LLQBranchManho les	Number LLQ Branch Manholes		= NumberLLQbranchcables * IF (LLQBranchBackboneUGDistance < 1, 0, LLQBranchBackboneUGDistance / VLOOKUP (Density, DistrSpacingTable, 2))	
AN	LLQStandardBran chManholeCost	LLQ Standard Branch Manhole Cost		= IF (OR (LLQBranchManholes = 0, LLQBranchDucts > 9) , 0, VLOOKUP (LLQBranchDucts, CHOOSE (FiberDepthCondition, ManholeHardRockTable, ManholeSoftRockTable, ManholeNormalSurfaceTable) , DensityColumnPointer))	
AO	LLQExpandedBra nchManholeCost	LLQ Expanded Branch Manhole Cost		= IF (OR (LLQBranchManholes = 0, LLQBranchDucts < = 9) , 0, VLOOKUP (9, CHOOSE (FiberDepthCondition, ManholeHardRockTable, ManholeSoftRockTable, ManholeNormalSurfaceTable) , DensityColumnPointer) + TRUNC (LLQBranchDucts / 9) * VLOOKUP (99, CHOOSE (FiberDepthCondition, ManholeHardRockTable, ManholeSoftRockTable, ManholeNormalSurfaceTable) , DensityColumnPointer))	
AP	LLQBackboneDuc ts	Number LLQ Backbone Ducts		= LLQMaxSizeBackboneCables + 2	
AQ	LLQBackboneMan holes	Number LLQ Backbone Manholes		= IF (LLQBranchBackboneUGDistance < 1, 0, LLQBranchBackboneUGDistance / 2 / VLOOKUP (Density, DistrSpacingTable, 2))	
AR	LLQStandardBack boneManholeCost	LLQ Standard Backbone Manhole Cost		= IF (OR (LLQBackboneManholes = 0, LLQBackboneDucts > 9) , 0, VLOOKUP (LLQBackboneDucts, CHOOSE (FiberDepthCondition, ManholeHardRockTable, ManholeSoftRockTable, ManholeNormalSurfaceTable) , DensityColumnPointer))	
AS	LLQExpandedBac kboneManholeCost	LLQ Expanded Backbone Manhole Cost		= IF (OR (LLQBackboneManholes = 0, LLQBackboneDucts < = 9) , 0, VLOOKUP (9, CHOOSE (FiberDepthCondition, ManholeHardRockTable, ManholeSoftRockTable, ManholeNormalSurfaceTable) , DensityColumnPointer) + TRUNC (LLQBackboneDucts / 9) * VLOOKUP (99, CHOOSE (FiberDepthCondition, ManholeHardRockTable, ManholeSoftRockTable, ManholeNormalSurfaceTable) , DensityColumnPointer))	
AT	LLQUGStructureC ost	LLQ Underground Structure Cost		= LLQBranchManholes * IF (LLQBranchDucts < = 9, LLQStandardBranchManholeCost, LLQExpandedBranchManholeCost) + LLQBackboneManholes * IF (LLQBackboneDucts < = 9, LLQStandardBackboneManholeCost, LLQExpandedBackboneManholeCost) + (NumberLLQbranchcables * LLQBranchDucts + LLQBackboneDucts) * LLQBranchBackboneUGDistance * VLOOKUP (0, ConduitDensityTable, DensityColumnPointer) + (NumberLLQbranchcables + 1 / 2) * LLQBranchBackboneUGDistance * VLOOKUP (Density, CHOOSE (FiberDepthCondition, HardRockStructure, SoftRockStructure, NormalStructure) , 3)	
AU	PairsperURQbranc hcable	Pairs Per URQ Branch Cable		= IF (NumberURQbranchcables = 0, 0, CEILING (NumberpairsinURquadrant / (2 * NumberURQbranchcables) / DistributionFillFactor, 1))	
AV	URQMaxSizeBran chCables	URQ Max Size Branch Cables		= IF (PairsperURQbranchcable > MaxDistrSize, TRUNC (PairsperURQbranchcable / MaxDistrSize) , 0)	

AW	URQResidualBranchCableSize	URQ Residual Branch Cable Size		= IF (PairsperURQbranchcable = 0, 0, INDEX (DistrCableSize, MATCH (PairsperURQbranchcable - URQMaxSizeBranchCables * MaxDistrSize, DistrCableSize, - 1) , 1))	
AX	URQMaxSizeBackboneCables	URQ Max Size Backbone Cables		= NumberURQbranchcables * URQMaxSizeBranchCables + IF (NumberURQbranchcables * URQResidualBranchCableSize > MaxDistrSize, TRUNC (NumberURQbranchcables * URQResidualBranchCableSize / MaxDistrSize) , 0)	
AY	URQResidualBackboneCableSize	URQ Residual Backbone Cable Size		= IF (URQResidualBranchCableSize = 0, 0, INDEX (DistrCableSize, MATCH (IF (NumberURQbranchcables * URQResidualBranchCableSize > MaxDistrSize, NumberURQbranchcables * URQResidualBranchCableSize - MaxDistrSize * TRUNC (NumberURQbranchcables * URQResidualBranchCableSize / MaxDistrSize) , NumberURQbranchcables * URQResidualBranchCableSize) , DistrCableSize, - 1) , 1))	
AZ	URQBranchBackboneAerialDistance	URQ Branch & Backbone Aerial Distance		= VLOOKUP (Density, CHOOSE (CopperDepthCondition, DistrHardMixTable, DistrSoftMixTable, DistrNormMixTable) , 4) * URQBackboneBranchCableLength	
BA	URQBranchAerialCableMaterial	URQ Branch Aerial Cable Material $		= IF (URQResidualBranchCableSize = 0, 0, NumberURQbranchcables * URQBranchBackboneAerialDistance * WaterCostMultiplier * CopperCostRatio * (VLOOKUP (URQResidualBranchCableSize, IF (URQCableGauge = 26, AERCopper26Table, AERCopper24Table) , DensityColumnPointer, FALSE) + URQMaxSizeBranchCables * VLOOKUP (MaxDistrSize, IF (URQCableGauge = 26, AERCopper26Table, AERCopper24Table) , DensityColumnPointer, FALSE)))	
BB	URQBackboneAerialCableMaterial	URQ Backbone Aerial Cable Material $		= IF (URQResidualBackboneCableSize = 0, 0, URQBranchBackboneAerialDistance * WaterCostMultiplier * CopperCostRatio * (VLOOKUP (URQResidualBackboneCableSize, IF (URQCableGauge = 26, AERCopper26Table, AERCopper24Table) , DensityColumnPointer, FALSE) + URQMaxSizeBackboneCables * VLOOKUP (MaxFeederSize, IF (URQCableGauge = 26, AERCopper26Table, AERCopper24Table) , DensityColumnPointer, FALSE)))	
BC	URQBranchBackboneBuriedDistance	URQ Branch & Backbone Buried Distance		= VLOOKUP (Density, CHOOSE (CopperDepthCondition, DistrHardMixTable, DistrSoftMixTable, DistrNormMixTable) , 3) * URQBackboneBranchCableLength	
BD	URQBranchBuriedCableMaterial	URQ Branch Buried Cable Material $		= IF (URQResidualBranchCableSize = 0, 0, NumberURQbranchcables * URQBranchBackboneBuriedDistance * WaterCostMultiplier * CopperCostRatio * (VLOOKUP (URQResidualBranchCableSize, IF (URQCableGauge = 26, BURCopper26Table, BURCopper24Table) , DensityColumnPointer, FALSE) + URQMaxSizeBranchCables * VLOOKUP (MaxDistrSize, IF (URQCableGauge = 26, BURCopper26Table, BURCopper24Table) , DensityColumnPointer, FALSE)))	
BE	URQBackboneBuriedCableMaterial	URQ Backbone Buried Cable Material $		= IF (URQResidualBackboneCableSize = 0, 0, URQBranchBackboneBuriedDistance * WaterCostMultiplier * CopperCostRatio * (VLOOKUP (URQResidualBackboneCableSize, IF (URQCableGauge = 26, BURCopper26Table, BURCopper24Table) , DensityColumnPointer, FALSE) + URQMaxSizeBackboneCables * VLOOKUP (MaxDistrSize, IF (URQCableGauge = 26, BURCopper26Table, BURCopper24Table) , DensityColumnPointer, FALSE)))	
BF	URQBranchBackboneUGDistance	URQ Branch & Backbone Underground Distance		= VLOOKUP (Density, CHOOSE (CopperDepthCondition, DistrHardMixTable, DistrSoftMixTable, DistrNormMixTable) , 2) * URQBackboneBranchCableLength	
BG	URQBranchUGCableMaterial	URQ Branch Underground Cable Material $		= IF (URQResidualBranchCableSize = 0, 0, NumberURQbranchcables * URQBranchBackboneUGDistance * WaterCostMultiplier * CopperCostRatio * (VLOOKUP (URQResidualBranchCableSize, IF (URQCableGauge = 26, UGCopper26Table, UGCopper24Table) , DensityColumnPointer, FALSE) + URQMaxSizeBranchCables * VLOOKUP (MaxDistrSize, IF (URQCableGauge = 26, UGCopper26Table, UGCopper24Table) , DensityColumnPointer, FALSE)))	
BH	URQBackboneUGCableMaterial	URQ Backbone Underground Cable Material $		= IF (URQResidualBackboneCableSize = 0, 0, URQBranchBackboneUGDistance * WaterCostMultiplier * CopperCostRatio * (VLOOKUP (URQResidualBackboneCableSize, IF (URQCableGauge = 26, UGCopper26Table, UGCopper24Table) , DensityColumnPointer, FALSE) + URQMaxSizeBackboneCables * VLOOKUP (MaxDistrSize, IF (URQCableGauge = 26, UGCopper26Table, UGCopper24Table) , DensityColumnPointer, FALSE)))	
BI	URQBranchDucts	Number URQ Branch Ducts		= URQMaxSizeBranchCables + 2	
BJ	URQBranchManholes	Number URQ Branch Manholes		= NumberURQbranchcables * IF (URQBranchBackboneUGDistance < 1, 0, URQBranchBackboneUGDistance / VLOOKUP (Density, DistrSpacingTable, 2))	
BK	URQStandardBranchManholeCost	URQ Standard Branch Manhole Cost		= IF (OR (URQBranchManholes = 0, URQBranchDucts > 9) , 0, VLOOKUP (URQBranchDucts, CHOOSE (FiberDepthCondition, ManholeHardRockTable, ManholeSoftRockTable, ManholeNormalSurfaceTable) , DensityColumnPointer))	

BL	URQExpandedBranchManholeCost	URQ Expanded Branch Manhole Cost		= IF (OR (URQBranchManholes = 0, URQBranchDucts < = 9) , 0, VLOOKUP (9, CHOOSE (FiberDepthCondition, ManholeHardRockTable, ManholeSoftRockTable, ManholeNormalSurfaceTable) , DensityColumnPointer) + TRUNC (URQBranchDucts / 9) * VLOOKUP (99, CHOOSE (FiberDepthCondition, ManholeHardRockTable, ManholeSoftRockTable, ManholeNormalSurfaceTable) , DensityColumnPointer))	
BM	URQBackboneDucts	Number URQ Backbone Ducts		= URQMaxSizeBackboneCables + 2	
BN	URQBackboneManholes	Number URQ Backbone Manholes		= IF (URQBranchBackboneUGDistance < 1, 0, URQBranchBackboneUGDistance / 2 / VLOOKUP (Density, DistrSpacingTable, 2))	
BO	URQStandardBackboneManholeCost	URQ Standard Backbone Manhole Cost		= IF (OR (URQBackboneManholes = 0, URQBackboneDucts > 9) , 0, VLOOKUP (URQBackboneDucts, CHOOSE (FiberDepthCondition, ManholeHardRockTable, ManholeSoftRockTable, ManholeNormalSurfaceTable) , DensityColumnPointer))	
BP	URQExpandedBackboneManholeCost	URQ Expanded Backbone Manhole Cost		= IF (OR (URQBackboneManholes = 0, URQBackboneDucts < = 9) , 0, VLOOKUP (9, CHOOSE (FiberDepthCondition, ManholeHardRockTable, ManholeSoftRockTable, ManholeNormalSurfaceTable) , DensityColumnPointer) + TRUNC (URQBackboneDucts / 9) * VLOOKUP (99, CHOOSE (FiberDepthCondition, ManholeHardRockTable, ManholeSoftRockTable, ManholeNormalSurfaceTable) , DensityColumnPointer))	
BQ	URQUGStructure Cost	URQ Underground Structure Cost		= URQBranchManholes * IF (URQBranchDucts < = 9, URQStandardBranchManholeCost, URQExpandedBranchManholeCost) + URQBackboneManholes * IF (URQBackboneDucts < = 9, URQStandardBackboneManholeCost, URQExpandedBackboneManholeCost) + (NumberURQbranchcables * URQBranchDucts + URQBackboneDucts) * URQBranchBackboneUGDistance * VLOOKUP (0, ConduitDensityTable, DensityColumnPointer) + (NumberURQbranchcables + 1 / 2) * URQBranchBackboneUGDistance * VLOOKUP (Density, CHOOSE (FiberDepthCondition, HardRockStructure, SoftRockStructure, NormalStructure) , 3)	
BR	PairsperLRQbranchcable	Pairs Per LRQ Branch Cable		= IF (NumberLRQbranchcables = 0, 0, CEILING (NumberpairsinLRquadrant / (2 * NumberLRQbranchcables) / DistributionFillFactor, 1))	
BS	LRQMaxSizeBranchCables	LRQ Max Size Branch Cables		= IF (PairsperLRQbranchcable > MaxDistrSize, TRUNC (PairsperLRQbranchcable / MaxDistrSize) , 0)	
BT	LRQResidualBranchCableSize	LRQ Residual Branch Cable Size		= IF (PairsperLRQbranchcable = 0, 0, INDEX (DistrCableSize, MATCH (PairsperLRQbranchcable - LRQMaxSizeBranchCables * MaxDistrSize, DistrCableSize, - 1) , 1))	
BU	LRQMaxSizeBackboneCables	LRQ Max Size Backbone Cables		= NumberLRQbranchcables * LRQMaxSizeBranchCables + IF (NumberLRQbranchcables * LRQResidualBranchCableSize > MaxDistrSize, TRUNC (NumberLRQbranchcables * LRQResidualBranchCableSize / MaxDistrSize) , 0)	
BV	LRQResidualBackboneCableSize	LRQ Residual Backbone Cable Size		= IF (LRQResidualBranchCableSize = 0, 0, INDEX (DistrCableSize, MATCH (IF (NumberLRQbranchcables * LRQResidualBranchCableSize > MaxDistrSize, NumberLRQbranchcables * LRQResidualBranchCableSize - MaxDistrSize * TRUNC (NumberLRQbranchcables * LRQResidualBranchCableSize / MaxDistrSize) , NumberLRQbranchcables * LRQResidualBranchCableSize) , DistrCableSize, - 1) , 1))	
BW	LRQBranchBackboneAerialDistance	LRQ Branch & Backbone Aerial Distance		= VLOOKUP (Density, CHOOSE (CopperDepthCondition, DistrHardMixTable, DistrSoftMixTable, DistrNormMixTable) , 4) * LRQBackboneBranchCableLength	
BX	LRQBranchAerialCableMaterial	LRQ Branch Aerial Cable Material $		= IF (LRQResidualBranchCableSize = 0, 0, NumberLRQbranchcables * LRQBranchBackboneAerialDistance * WaterCostMultiplier * CopperCostRatio * (VLOOKUP (LRQResidualBranchCableSize, IF (LRQCableGauge = 26, AERCopper26Table, AERCopper24Table) , DensityColumnPointer, FALSE) + LRQMaxSizeBranchCables * VLOOKUP (MaxDistrSize, IF (LRQCableGauge = 26, AERCopper26Table, AERCopper24Table) , DensityColumnPointer, FALSE)))	
BY	LRQBackboneAerialCableMaterial	LRQ Backbone Aerial Cable Material $		= IF (LRQResidualBackboneCableSize = 0, 0, LRQBranchBackboneAerialDistance * WaterCostMultiplier * CopperCostRatio * (VLOOKUP (LRQResidualBackboneCableSize, IF (LRQCableGauge = 26, AERCopper26Table, AERCopper24Table) , DensityColumnPointer, FALSE) + LRQMaxSizeBackboneCables * VLOOKUP (MaxFeederSize, IF (LRQCableGauge = 26, AERCopper26Table, AERCopper24Table) , DensityColumnPointer, FALSE)))	
BZ	LRQBranchBackboneBuriedDistance	LRQ Branch & Backbone Buried Distance		= VLOOKUP (Density, CHOOSE (CopperDepthCondition, DistrHardMixTable, DistrSoftMixTable, DistrNormMixTable) , 3) * LRQBackboneBranchCableLength	
CA	LRQBranchBuriedCableMaterial	LRQ Branch Buried Cable Material $		= IF (LRQResidualBranchCableSize = 0, 0, NumberLRQbranchcables * LRQBranchBackboneBuriedDistance * WaterCostMultiplier * CopperCostRatio * (VLOOKUP (LRQResidualBranchCableSize, IF (LRQCableGauge = 26, BURCopper26Table, BURCopper24Table) , DensityColumnPointer, FALSE) + LRQMaxSizeBranchCables * VLOOKUP (MaxDistrSize, IF (LRQCableGauge = 26, BURCopper26Table, BURCopper24Table) , DensityColumnPointer, FALSE)))	

CB	LRQBackboneBuriedCableMaterial	LRQ Backbone Buried Cable Material $		= IF (LRQResidualBackboneCableSize = 0, 0, LRQBranchBackboneBuriedDistance * WaterCostMultiplier * CopperCostRatio * (VLOOKUP (LRQResidualBackboneCableSize, IF (LRQCableGauge = 26, BURCopper26Table, BURCopper24Table) , DensityColumnPointer, FALSE) + LRQMaxSizeBackboneCables * VLOOKUP (MaxDistrSize, IF (LRQCableGauge = 26, BURCopper26Table, BURCopper24Table) , DensityColumnPointer, FALSE)))	
CC	LRQBranchBackboneUGDistance	LRQ Branch & Backbone Underground Distance		= VLOOKUP (Density, CHOOSE (CopperDepthCondition, DistrHardMixTable, DistrSoftMixTable, DistrNormMixTable) , 2) * LRQBackboneBranchCableLength	
CD	LRQBranchUGCableMaterial	LRQ Branch Underground Cable Material $		= IF (LRQResidualBranchCableSize = 0, 0, NumberLRQbranchcables * LRQBranchBackboneUGDistance * WaterCostMultiplier * CopperCostRatio * (VLOOKUP (LRQResidualBranchCableSize, IF (LRQCableGauge = 26, UGCopper26Table, UGCopper24Table) , DensityColumnPointer, FALSE) + LRQMaxSizeBranchCables * VLOOKUP (MaxDistrSize, IF (LRQCableGauge = 26, UGCopper26Table, UGCopper24Table) , DensityColumnPointer, FALSE)))	
CE	LRQBackboneUGCableMaterial	LRQ Backbone Underground Cable Material $		= IF (LRQResidualBackboneCableSize = 0, 0, LRQBranchBackboneUGDistance * WaterCostMultiplier * CopperCostRatio * (VLOOKUP (LRQResidualBackboneCableSize, IF (LRQCableGauge = 26, UGCopper26Table, UGCopper24Table) , DensityColumnPointer, FALSE) + LRQMaxSizeBackboneCables * VLOOKUP (MaxDistrSize, IF (LRQCableGauge = 26, UGCopper26Table, UGCopper24Table) , DensityColumnPointer, FALSE)))	
CF	LRQBranchDucts	Number LRQ Branch Ducts		= LRQMaxSizeBranchCables + 2	
CG	LRQBranchManholes	Number LRQ Branch Manholes		= NumberLRQbranchcables * IF (LRQBranchBackboneUGDistance < 1, 0, LRQBranchBackboneUGDistance / VLOOKUP (Density, DistrSpacingTable, 2))	
CH	LRQStandardBranchManholeCost	LRQ Standard Branch Manhole Cost		= IF (OR (LRQBranchManholes = 0, LRQBranchDucts > 9) , 0, VLOOKUP (LRQBranchDucts, CHOOSE (FiberDepthCondition, ManholeHardRockTable, ManholeSoftRockTable, ManholeNormalSurfaceTable) , DensityColumnPointer))	
CI	LRQExpandedBranchManholeCost	LRQ Expanded Branch Manhole Cost		= IF (OR (LRQBranchManholes = 0, LRQBranchDucts < = 9) , 0, VLOOKUP (9, CHOOSE (FiberDepthCondition, ManholeHardRockTable, ManholeSoftRockTable, ManholeNormalSurfaceTable) , DensityColumnPointer) + TRUNC (LRQBranchDucts / 9) * VLOOKUP (99, CHOOSE (FiberDepthCondition, ManholeHardRockTable, ManholeSoftRockTable, ManholeNormalSurfaceTable) , DensityColumnPointer))	
CJ	LRQBackboneDucts	Number LRQ Backbone Ducts		= LRQMaxSizeBackboneCables + 2	
CK	LRQBackboneManholes	Number LRQ Backbone Manholes		= IF (LRQBranchBackboneUGDistance < 1, 0, LRQBranchBackboneUGDistance / 2 / VLOOKUP (Density, DistrSpacingTable, 2))	
CL	LRQStandardBackboneManholeCost	LRQ Standard Backbone Manhole Cost		= IF (OR (LRQBackboneManholes = 0, LRQBackboneDucts > 9) , 0, VLOOKUP (LRQBackboneDucts, CHOOSE (FiberDepthCondition, ManholeHardRockTable, ManholeSoftRockTable, ManholeNormalSurfaceTable) , DensityColumnPointer))	
CM	LRQExpandedBackboneManholeCost	LRQ Expanded Backbone Manhole Cost		= IF (OR (LRQBackboneManholes = 0, LRQBackboneDucts < = 9) , 0, VLOOKUP (9, CHOOSE (FiberDepthCondition, ManholeHardRockTable, ManholeSoftRockTable, ManholeNormalSurfaceTable) , DensityColumnPointer) + TRUNC (LRQBackboneDucts / 9) * VLOOKUP (99, CHOOSE (FiberDepthCondition, ManholeHardRockTable, ManholeSoftRockTable, ManholeNormalSurfaceTable) , DensityColumnPointer))	
CN	LRQUGStructureCost	LRQ Underground Structure Cost		= LRQBranchManholes * IF (LRQBranchDucts < = 9, LRQStandardBranchManholeCost, LRQExpandedBranchManholeCost) + LRQBackboneManholes * IF (LRQBackboneDucts < = 9, LRQStandardBackboneManholeCost, LRQExpandedBackboneManholeCost) + (NumberLRQbranchcables * LRQBranchDucts + LRQBackboneDucts) * LRQBranchBackboneUGDistance * VLOOKUP (0, ConduitDensityTable, DensityColumnPointer) + (NumberLRQbranchcables + 1 / 2) * LRQBranchBackboneUGDistance * VLOOKUP (Density, CHOOSE (FiberDepthCondition, HardRockStructure, SoftRockStructure, NormalStructure) , 3)	
CO	SubtotalAerialCableDistributionCost	Subtotal Aerial Cable Distribution Cost		= ULQBranchAerialCableMaterial + ULQBackboneAerialCableMaterial + LLQBranchAerialCableMaterial + LLQBackboneAerialCableMaterial + URQBranchAerialCableMaterial + URQBackboneAerialCableMaterial + LRQBranchAerialCableMaterial + LRQBackboneAerialCableMaterial	
CP	SubtotalBuriedCableDistributionCost	Subtotal Buried Cable Distribution Cost		= ULQBranchBuriedCableMaterial + ULQBackboneBuriedCableMaterial + LLQBranchBuriedCableMaterial + LLQBackboneBuriedCableMaterial + URQBranchBuriedCableMaterial + URQBackboneBuriedCableMaterial + LRQBranchBuriedCableMaterial + LRQBackboneBuriedCableMaterial	

CQ	TotalUGCableDistributionCost	Total Underground Cable Distribution Cost		= ULQBranchUGCableMaterial + ULQBackboneUGCableMaterial + LLQBranchUGCableMaterial + LLQBackboneUGCableMaterial + URQBranchUGCableMaterial + URQBackboneUGCableMaterial + LRQBranchUGCableMaterial + LRQBackboneUGCableMaterial	
CR	TotalAerialCableDistributionDistance	Total Aerial Cable Distribution Distance		= ULQBranchBackboneAerialDistance * (NumberULQbranchcables + 1 / 2) + LLQBranchBackboneAerialDistance * (NumberLLQbranchcables + 1 / 2) + URQBranchBackboneAerialDistance * (NumberURQbranchcables + 1 / 2) + LRQBranchBackboneAerialDistance * (NumberLRQbranchcables + 1 / 2)	
CS	NumberPolesforDistribution	Number Of Poles for Distribution		= TotalAerialCableDistributionDistance / VLOOKUP (Density, DistrSpacingTable, 3)	
CT	TotalPoleDistributionCost	Total Pole Distribution Cost		= NumberPolesForDistribution * WaterCostMultiplier * VLOOKUP (Density, CHOOSE (CopperDepthCondition, HardRockStructure, SoftRockStructure, NormalStructure) , 7)	
CU	TotalBuriedCableDistributionDistance	Total Buried Cable Distribution Distance		= ULQBranchBackboneBuriedDistance * (NumberULQbranchcables + 1 / 2) + LLQBranchBackboneBuriedDistance * (NumberLLQbranchcables + 1 / 2) + URQBranchBackboneBuriedDistance * (NumberURQbranchcables + 1 / 2) + LRQBranchBackboneBuriedDistance * (NumberLRQbranchcables + 1 / 2)	
CV	TotalBuriedDistributionStructureCost	Total Buried Distribution Structure Cost		= IF (TotalBuriedCableDistributionDistance < 1, 0, WaterCostMultiplier * TotalBuriedCableDistributionDistance * VLOOKUP (Density, CHOOSE (CopperDepthCondition, HardRockStructure, SoftRockStructure, NormalStructure) , 5))	
CW	TotalDistributionConduitCost	Total Distribution Conduit Cost		= ULQUGStructureCost + LLQUGStructureCost + URQUGStructureCost + LRQUGStructureCost	

Sheet: Term,Drop,&NID

Col	Range Name	Column Name	Column Comment	Formula	Formula Comment
A	PercentDropsAerial	Percent Drops Aerial		= VLOOKUP (Density, CHOOSE (CopperDepthCondition, CopperHardMixTable, CopperSoftMixTable, CopperNormMixTable) , 4)	
B	CostperRESNID	Cost Per Res NID		= VLOOKUP ("NID", RESNIDTABLE, DensityColumnPointer, FALSE) + (PairsPerHousingUnit + 1) * (VLOOKUP ("PROTECTOR", RESNIDTABLE, DensityColumnPointer, FALSE) + VLOOKUP ("INTERFACE", RESNIDTABLE, DensityColumnPointer, FALSE))	
C	CostperBUSNID	Cost Per Bus NID		= VLOOKUP ("NID", BUSNIDTABLE, DensityColumnPointer, FALSE) + PairsPerBusinessLocation * (VLOOKUP ("PROTECTOR", BUSNIDTABLE, DensityColumnPointer, FALSE) + VLOOKUP ("INTERFACE", BUSNIDTABLE, DensityColumnPointer, FALSE))	
D	AerialDropCostperFoot	Aerial Drop Cost Per Foot		= VLOOKUP (1, AerDropCost, DensityColumnPointer)	
E	BuriedDropCostperFoot	Buried Drop Cost Per Foot		= VLOOKUP (1, BurDropCost, DensityColumnPointer)	
F	_1to2HUperDwellingGridLocations	1&2 HU Per Dwelling Grid Drops		= SingleUnitDetachedHUlocations + SingleUnitAttachedHUlocations + 2 * _1to2Buildings + MobileHUlocations + otherBuildings	
G	PercentGT2MultiFamilyLots	Percent >2 Multi-Family Lots		= IF (TotalHousingUnitLocations = 0, 0, (TotalHousingUnitLocations - (SingleUnitDetachedHUlocations + SingleUnitAttachedHUlocations + _1to2Buildings + MobileHUlocations + otherBuildings)) / TotalHousingUnitLocations)	
H	ULQTerminals	ULQ Terminals		= IF (OR (ULQLots = "", ULQLots = 0) , 0, IF (NumberULQbranchcables = 0, 1, NumberULQbranchcables^2))	
I	ULQDropLength	ULQ Drop Length		= MIN (500, ULQLotSideLength * 0.5 * SQRT (2))	
J	NumberULQResDrops	Number Of ULQ Res Drops		= IF (TotalHousingUnitLocations = 0, 0, (ULQHousingUnitslocations / TotalHousingUnitLocations) * _1to2HUperDwellingGridLocations)	
K	AvgResDropsperULQterminal	Avg Res Drops Per ULQ Terminal		= IF (ULQTerminals = 0, 0, CEILING (NumberULQResDrops / ULQTerminals, 1))	
L	SmallBusCheck	Small Business Line Per Location Check		= IF (BusinessLocations = 0, 0, IF (TotalBusinessLines / BusinessLocations > 6, 0, 1))	
M	AvgBUSDropsperULQTerminal	Avg Bus Drops Per ULQ Terminal		= IF (ULQTerminals = 0, 0, CEILING (ULQBusinesslocations / ULQTerminals, 1)) * SmallBusCheck	
N	ULQDropTerminalPairs	ULQ Drop Terminal Pairs		= AvgResDropsperULQterminal * PairsPerHousingUnit + AvgBUSDropsperULQTerminal * PairsPerBusinessLocation	
O	TotalULQAerialDropTerminalCost	Total ULQ Aerial Drop Terminal Cost		= IF (ULQDropTerminalPairs = 0, 0, ULQTerminals * PercentDropsAerial * VLOOKUP (ULQDropTerminalPairs, AerTermCostTable, DensityColumnPointer))	
P	TotalULQBuriedDropTerminalCost	Total ULQ Buried Drop Terminal Cost		= IF (ULQDropTerminalPairs = 0, 0, ULQTerminals * (1 - PercentDropsAerial) * VLOOKUP (ULQDropTerminalPairs, BURTermCostTable, DensityColumnPointer))	
Q	TotalULQNIDCost	Total ULQ NID Cost		= IF (NumberULQResDrops = 0, 0, NumberULQResDrops * CostperResNID) + IF (ULQBusinesslocations = 0, 0, ULQBusinesslocations * CostperBusNID)	
R	ULQAerialDropCost	ULQ Aerial Drop Cost		= ULQDropLength * (NumberULQResDrops + ULQBusinesslocations) * PercentDropsAerial * AerialDropCostperFoot	
S	ULQBuriedDropCost	ULQ Buried Drop Cost		= ULQDropLength * (NumberULQResDrops + ULQBusinesslocations) * (1 - PercentDropsAerial) * BuriedDropCostperFoot	
T	LLQTerminals	LLQ Terminals		= IF (OR (LLQLots = "", LLQLots = 0) , 0, IF (NumberLLQbranchcables = 0, 1, NumberLLQbranchcables^2))	
U	LLQDropLength	LLQ Drop Length		= IF (LLQLotSideLength * 0.5 * SQRT (2) > 500, 500, LLQLotSideLength * 0.5 * SQRT (2))	
V	NumberLLQRESDrops	Number Of LLQ Res Drops		= IF (TotalHousingUnitLocations = 0, 0, (LLQHousingUnitlocations / TotalHousingUnitLocations) * _1to2HUperDwellingGridLocations)	
W	AvgRESDropsperLLQterminal	Avg Res Drops Per LLQ Terminal		= IF (LLQTerminals = 0, 0, CEILING (NumberLLQResDrops / LLQTerminals, 1))	

X	AvgBUSDropsper LLQTerminal	Avg Bus Drops Per LLQ Terminal		= IF (LLQTerminals = 0, 0, CEILING (LLQBusinesslocations / LLQTerminals, 1)) * SmallBusCheck	
Y	LLQDropTerminal Pairs	LLQ Drop Terminal Pairs		= AvgRESDropsperLLQterminal * PairsPerHousingUnit + AvgBUSDropsPerLLQTerminal * PairsPerBusinessLocation	
Z	TotalLLQAerialDr opTerminalCost	Total LLQ Aerial Drop Terminal Cost		= IF (LLQDropTerminalPairs = 0, 0, LLQTerminals * PercentDropsAerial * VLOOKUP (LLQDropTerminalPairs, AerTermCostTable, DensityColumnPointer))	
AA	TotalLLQBuriedD ropTerminalCost	Total LLQ Buried Drop Terminal Cost		= IF (LLQDropTerminalPairs = 0, 0, LLQTerminals * (1 - PercentDropsAerial) * VLOOKUP (LLQDropTerminalPairs, BURTermCostTable, DensityColumnPointer))	
AB	TotalLLQNIDCost	Total LLQ NID Cost		= IF (NumberLLQResDrops = 0, 0, NumberLLQResDrops * CostperResNID)	
AC	LLQAerialDropCo st	LLQ Aerial Drop Cost		= LLQDropLength * (NumberLLQResDrops + LLQBusinesslocations) * PercentDropsAerial * AerialDropCostperFoot	
AD	LLQBuriedDropC ost	LLQ Buried Drop Cost		= LLQDropLength * (NumberLLQResDrops + LLQBusinesslocations) * (1 - PercentDropsAerial) * BuriedDropCostperFoot	
AE	URQTerminals	URQ Terminals		= IF (OR (URQLots = "", URQLots = 0) , 0, IF (NumberURQbranchcables = 0, 1, NumberURQbranchcables^2))	
AF	URQDropLength	URQ Drop Length		= IF (URQLotSideLength * 0.5 * SQRT (2) > 500, 500, URQLotSideLength * 0.5 * SQRT (2))	
AG	NumberURQRES Drops	Number Of URQ Res Drops		= IF (TotalHousingUnitLocations = 0, 0, (URQHousingUnitlocations / TotalHousingUnitLocations) * 1to2HUperDwellingGridLocations)	
AH	AvgRESDropsper URQterminal	Avg Res Drops Per URQ Terminal		= IF (URQTerminals = 0, 0, CEILING (NumberURQResDrops / URQTerminals, 1))	
AI	AvgBUSDropsper URQTerminal	Avg Bus Drops Per URQ Terminal		= IF (URQTerminals = 0, 0, CEILING (URQBusinesslocations / URQTerminals, 1)) * SmallBusCheck	
AJ	URQDropTermina lPairs	URQ Drop Terminal Pairs		= AvgRESDropsperURQterminal * PairsPerHousingUnit + AvgBUSDropsperURQTerminal * PairsPerBusinessLocation	
AK	TotalURQAerialDr opTerminalCost	Total URQ Aerial Drop Terminal Cost		= IF (URQDropTerminalPairs = 0, 0, URQTerminals * PercentDropsAerial * VLOOKUP (URQDropTerminalPairs, AerTermCostTable, DensityColumnPointer))	
AL	TotalURQBuriedD ropTerminalCost	Total URQ Buried Drop Terminal Cost		= IF (URQDropTerminalPairs = 0, 0, URQTerminals * (1 - PercentDropsAerial) * VLOOKUP (URQDropTerminalPairs, BURTermCostTable, DensityColumnPointer))	
AM	TotalURQNIDCos t	Total URQ NID Cost		= IF (NumberURQResDrops = 0, 0, NumberURQResDrops * CostperResNID) + IF (URQBusinesslocations = 0, 0, URQBusinesslocations * CostperBusNID)	
AN	URQAerialDropCo st	URQ Aerial Drop Cost		= URQDropLength * (NumberURQResDrops + URQBusinesslocations) * PercentDropsAerial * AerialDropCostperFoot	
AO	URQBuriedDropC ost	URQ Buried Drop Cost		= URQDropLength * (NumberURQResDrops + URQBusinesslocations) * (1 - PercentDropsAerial) * BuriedDropCostperFoot	
AP	LRQTerminals	LRQ Terminals		= IF (OR (LRQLots = "", LRQLots = 0) , 0, IF (NumberLRQbranchcables = 0, 1, NumberLRQbranchcables^2))	
AQ	LRQDropLength	LRQ Drop Length		= MIN (500, LRQLotSideLength * 0.5 * SQRT (2))	
AR	NumberLRQRES Drops	Number Of LRQ Res Drops		= IF (TotalHousingUnitLocations = 0, 0, (LRQHousingUnitlocations / TotalHousingUnitLocations) * 1to2HUperDwellingGridLocations)	
AS	AvgRESDropsperL RQterminal	Avg Res Drops Per LRQ Terminal		= IF (LRQTerminals = 0, 0, CEILING (NumberLRQResDrops / LRQTerminals, 1))	
AT	AvgBUSDropsper LRQTerminal	Avg Bus Drops Per LRQ Terminal		= IF (LRQTerminals = 0, 0, CEILING (LRQBusinesslocations / LRQTerminals, 1)) * SmallBusCheck	
AU	LRQDropTerminal Pairs	LRQ Drop Terminal Pairs		= AvgRESDropsperLRQterminal * PairsPerHousingUnit + AvgBUSDropsPerLRQTerminal * PairsPerBusinessLocation	
AV	TotalLRQAerialDr opTerminalCost	Total LRQ Aerial Drop Terminal Cost		= IF (LRQDropTerminalPairs = 0, 0, LRQTerminals * PercentDropsAerial * VLOOKUP (LRQDropTerminalPairs, AerTermCostTable, DensityColumnPointer))	

AW	TotalLRQBuriedDropTerminalCost	Total LRQ Buried Drop Terminal Cost		= IF (LRQDropTerminalPairs = 0, 0, LRQTerminals * (1 - PercentDropsAerial) * VLOOKUP (LRQDropTerminalPairs, BURTermCostTable, DensityColumnPointer))	
AX	TotalLRQNIDCost	Total LRQ NID Cost		= IF (NumberLRQResDrops = 0, 0, NumberLRQResDrops * CostperResNID) + IF (LRQBusinesslocations = 0, 0, LRQBusinesslocations * CostperBusNID)	
AY	LRQAerialDropCost	LRQ Aerial Drop Cost		= LRQDropLength * (NumberLRQResDrops + LRQBusinesslocations) * PercentDropsAerial * VLOOKUP (1, AerDropCost, DensityColumnPointer)	
AZ	LRQBuriedDropCost	LRQ Buried Drop Cost		= LRQDropLength * (NumberLRQResDrops + LRQBusinesslocations) * (1 - PercentDropsAerial) * VLOOKUP (1, BurDropCost, DensityColumnPointer)	
BA	TotalGridAerialDropTerminalCost	Total Grid Aerial Drop Terminal Cost		= TotalULQAerialDropTerminalCost + TotalLLQAerialDropTerminalCost + TotalURQAerialDropTerminalCost + TotalLRQAerialDropTerminalCost	
BB	TotalGridBuriedDropTerminalCost	Total Grid Buried Drop Terminal Cost		= TotalULQBuriedDropTerminalCost + TotalLLQBuriedDropTerminalCost + TotalURQBuriedDropTerminalCost + TotalLRQBuriedDropTerminalCost	
BC	TotalGridNIDCost	Total Grid NID Cost		= TotalULQNIDCost + TotalLLQNIDCost + TotalURQNIDCost + TotalLRQNIDCost	
BD	TotalGridSingleAerialDropCost	Total Grid Single Aerial Drop Cost		= ULQAerialDropCost + LLQAerialDropCost + URQAerialDropCost + LRQAerialDropCost	
BE	TotalGridSingleBuriedDropCost	Total Grid Single Buried Drop Cost		= ULQBuriedDropCost + LLQBuriedDropCost + URQBuriedDropCost + LRQBuriedDropCost	
BF	WeightedGridMultiHUDropLength	Weighted Grid Multi-HU Drop Length		= IF (OR (TotalHousingUnitLocations = 0, PercentGt2MultiFamilyLots = 0) , 0, (ULQHousingUnitslocations * ULQDropLength + LLQHousingUnitlocations * LLQDropLength + URQHousingUnitlocations * URQDropLength + LRQHousingUnitlocations * LRQDropLength) / TotalHousingUnitLocations)	
BG	CableSizeLargeBusLocations	Cable Size Large Business Locations		= IF (OR (SmallBusCheck = 1, BusinessLocations = 0) , 0, INDEX (DistrCableSize, MATCH (TotalBusinessLines / BusinessLocations, DistrCableSize, - 1) , 1))	
BH	AerialDropCostLargeBus	Aerial Drop Cable Cost Large Business Locations		= IF (CableSizeLargeBusLocations = 0, 0, BusinessLocations * PercentDropsAerial * WeightedGridMultiHUDropLength * VLOOKUP (CableSizeLargeBusLocations, AERCopper24Table, DensityColumnPointer, FALSE))	
BI	BuriedDropCostLargeBus	Buried Drop Cable Cost Large Business Locations		= IF (CableSizeLargeBusLocations = 0, 0, BusinessLocations * (1 - PercentDropsAerial) * WeightedGridMultiHUDropLength * VLOOKUP (CableSizeLargeBusLocations, BURCopper24Table, DensityColumnPointer, FALSE))	
BJ	CostPerTerminalLargeBus	Cost Per Terminal Large At Large Business Locations		= IF (CableSizeLargeBusLocations = 0, 0, VLOOKUP (CableSizeLargeBusLocations, IndoorSAICostTable, DensityColumnPointer))	
BK	CableSize3to4HUdwellings	Cable Size 3-4 HU Dwellings		= MAX (25, INDEX (DistrCableSize, MATCH (VLOOKUP ("3 - 4", HousingUnitsperDwelling, DensityColumnPointer, FALSE) * PairsPerHousingUnit, DistrCableSize, - 1) , 1))	
BL	AerialDropCableCost3to4HUDwellings	Aerial Drop Cable Cost 3-4 HU Dwellings		= _3to4Buildings * PercentDropsAerial * WeightedGridMultiHUDropLength * VLOOKUP (CableSize3to4HUdwellings, AERCopper24Table, DensityColumnPointer, FALSE)	
BM	BuriedDropCableCost3to4HUDwellings	Buried Drop Cable Cost 3-4 HU Dwellings		= _3to4Buildings * (1 - PercentDropsAerial) * WeightedGridMultiHUDropLength * VLOOKUP (CableSize3to4HUdwellings, BURCopper24Table, DensityColumnPointer, FALSE)	
BN	CostperTerminal3to4HUdwellings	Cost Per Terminal 3-4 HU Dwellings		= IF (CableSize3to4HUdwellings = 0, 0, VLOOKUP (CableSize3to4HUdwellings, IndoorSAICostTable, DensityColumnPointer))	
BO	CableSize5to9HUdwellings	Cable Size 5-9 HU Dwellings		= MAX (25, INDEX (DistrCableSize, MATCH (VLOOKUP ("5 - 9", HousingUnitsperDwelling, DensityColumnPointer, FALSE) * PairsPerHousingUnit, DistrCableSize, - 1) , 1))	
BP	AerialDropCableCost5to9HUDwellings	Aerial Drop Cable Cost 5-9 HU Dwellings		= _5to9Buildings * PercentDropsAerial * WeightedGridMultiHUDropLength * VLOOKUP (CableSize5to9HUdwellings, AERCopper24Table, DensityColumnPointer, FALSE)	

BQ	BuriedDropCableCost5to9HUDwellings	Buried Drop Cable Cost 5-9 HU Dwellings		= _5to9Buildings * (1 - PercentDropsAerial) * WeightedGridMultiHUDropLength * VLOOKUP (CableSize5to9HUdwellings, BURCopper24Table, DensityColumnPointer, FALSE)	
BR	CostperTerminal5to9HUdwellings	Cost Per Terminal 5-9 HU Dwellings		= IF (CableSize5to9HUdwellings = 0, 0, VLOOKUP (CableSize5to9HUdwellings, IndoorSAICostTable, DensityColumnPointer))	
BS	CableSize10to19HUdwellings	Cable Size 10-19 HU Dwellings		= MAX (25, INDEX (DistrCableSize, MATCH (VLOOKUP ("10 - 19", HousingUnitsperDwelling, DensityColumnPointer, FALSE) * PairsPerHousingUnit, DistrCableSize, - 1) , 1))	
BT	AerialDropCableCost10to19HUDwellings	Aerial Drop Cable Cost 10-19 HU Dwellings		= _10to19Buildings * PercentDropsAerial * WeightedGridMultiHUDropLength * VLOOKUP (CableSize10to19HUdwellings, AERCopper24Table, DensityColumnPointer, FALSE)	
BU	BuriedDropCableCost10to19HUDwellings	Buried Drop Cable Cost 10-19 HU Dwellings		= _10to19Buildings * (1 - PercentDropsAerial) * WeightedGridMultiHUDropLength * VLOOKUP (CableSize10to19HUdwellings, BURCopper24Table, DensityColumnPointer, FALSE)	
BV	CostperTerminal10to19HUdwellings	Cost Per Terminal 10-19 HU Dwellings		= IF (CableSize10to19HUdwellings = 0, 0, VLOOKUP (CableSize10to19HUdwellings, IndoorSAICostTable, DensityColumnPointer))	
BW	CableSize20to49HUdwellings	Cable Size 20-49 HU Dwellings		= MAX (50, INDEX (DistrCableSize, MATCH (VLOOKUP ("20 - 49", HousingUnitsperDwelling, DensityColumnPointer, FALSE) * PairsPerHousingUnit, DistrCableSize, - 1) , 1))	
BX	AerialDropCableCost20to49HUDwellings	Aerial Drop Cable Cost 20-49 HU Dwellings		= _20to49Buildings * PercentDropsAerial * WeightedGridMultiHUDropLength * VLOOKUP (CableSize20to49HUdwellings, AERCopper24Table, DensityColumnPointer, FALSE)	
BY	BuriedDropCableCost20to49HUDwellings	Buried Drop Cable Cost 20-49 HU Dwellings		= _20to49Buildings * (1 - PercentDropsAerial) * WeightedGridMultiHUDropLength * VLOOKUP (CableSize20to49HUdwellings, BURCopper24Table, DensityColumnPointer, FALSE)	
BZ	CostperTerminal20to49HUdwellings	Cost Per Terminal 20-49 HU Dwellings		= IF (CableSize20to49HUdwellings = 0, 0, VLOOKUP (CableSize20to49HUdwellings, IndoorSAICostTable, DensityColumnPointer))	
CA	CableSizeOver50HUdwellings	Cable Size >50 HU Dwellings		= MAX (100, INDEX (DistrCableSize, MATCH (VLOOKUP (" > 50", HousingUnitsperDwelling, DensityColumnPointer, FALSE) * PairsPerHousingUnit, DistrCableSize, - 1) , 1))	
CB	AerialDropCableCostOver50HUDwellings	Aerial Drop Cable Cost >50 HU Dwellings		= over50Buildings * PercentDropsAerial * WeightedGridMultiHUDropLength * VLOOKUP (CableSizeOver50HUdwellings, AERCopper24Table, DensityColumnPointer, FALSE)	
CC	BuriedDropCableCostOver50HUDwellings	Buried Drop Cable Cost >50 HU Dwellings		= over50Buildings * (1 - PercentDropsAerial) * WeightedGridMultiHUDropLength * VLOOKUP (CableSizeOver50HUdwellings, BURCopper24Table, DensityColumnPointer, FALSE)	
CD	CostperTerminalOver50HUdwellings	Cost Per Terminal >50 HU Dwellings		= IF (CableSizeOver50HUdwellings = 0, 0, VLOOKUP (CableSizeOver50HUdwellings, IndoorSAICostTable, DensityColumnPointer))	
CE	MultiHUAerialDropCable	Multi-HU Aerial Drop Cable		= AerialDropCableCost3to4HUDwellings + AerialDropCableCost5to9HUDwellings + AerialDropCableCost10to19HUDwellings + AerialDropCableCost20to49HUDwellings + AerialDropCableCostOver50HUDwellings + AerialDropCostLargeBus	
CF	MultiHUBuriedDropCable	Multi-HU Buried Drop Cable		= BuriedDropCableCost3to4HUDwellings + BuriedDropCableCost5to9HUDwellings + BuriedDropCableCost10to19HUDwellings + BuriedDropCableCost20to49HUDwellings + BuriedDropCableCostOver50HUDwellings + BuriedDropCostLargeBus	
CG	TotalMultiHUBuriedDropCableDistance	Total Multi-HU Buried Drop Cable Distance		= (_3to4Buildings + _5to9Buildings + _10to19Buildings + _20to49Buildings + over50Buildings + BusinessLocations * (1 - SmallBusCheck)) * WeightedGridMultiHUDropLength	
CH	PlacementCostMHUBuriedDropCable	Placement Cost MHU Buried Drop Cable		= IF (TotalMultiHUBuriedDropCableDistance < 1, 0, TotalMultiHUBuriedDropCableDistance * VLOOKUP (Density, CHOOSE (CopperDepthCondition, HardRockStructure, SoftRockStructure, NormalStructure) , 4))	

CI	MultiHUAerialBuildingTerminalCost	Multi-HU Aerial Building Terminal Cost		= (_3to4Buildings * CostperTerminal3to4HUdwellings + _5to9Buildings * CostperTerminal5to9HUdwellings + _10to19Buildings * CostperTerminal10to19HUdwellings + _20to49Buildings * CostperTerminal20to49HUdwellings + over50Buildings * CostperTerminalOver50HUdwellings + BusinessLocations * CostPerTerminalLargeBus) * PercentDropsAerial	
CJ	MultiHUBuriedBuildingTerminalCost	Multi-HU Buried Building Terminal Cost		= (_3to4Buildings * CostperTerminal3to4HUdwellings + _5to9Buildings * CostperTerminal5to9HUdwellings + _10to19Buildings * CostperTerminal10to19HUdwellings + _20to49Buildings * CostperTerminal20to49HUdwellings + over50Buildings * CostperTerminalOver50HUdwellings + BusinessLocations * CostPerTerminalLargeBus) * (1 - PercentDropsAerial)	
CK	SubtotalAerialCableRelatedcosts	Subtotal Aerial Cable Related costs		= TotalGridAerialDropTerminalCost + TotalGridSingleAerialDropCost + MultiHUAerialDropCable + MultiHUAerialBuildingTerminalCost	
CL	SubtotalBuriedCableRelatedcosts	Subtotal Buried Cable Related costs		= TotalGridBuriedDropTerminalCost + TotalGridSingleBuriedDropCost + MultiHUBuriedDropCable + MultiHUBuriedBuildingTerminalCost + PlacementCostMHUBuriedDropCable	
CM	TotalAerialCableDistributionCost	Total Aerial Cable Distribution Cost		= SubtotalAerialCableDistributionCost + SubtotalAerialCablerelatedcosts	
CN	TotalBuriedCable DistributionCost	Total Buried Cable Distribution Cost		= SubtotalBuriedCableDistributionCost + TotalBuriedDistributionStructureCost + SubtotalBuriedCablerelatedcosts	
CO		Total UG Cable Distribution Cost		= TotalUGCableDistributionCost	
CP		Total Pole Distribution Cost		= TotalPoleDistributionCost	
CQ		Total Conduit Distribution Cost		= TotalDistributionConduitCost	
CR	TotalNIDCostAllocatedAerial	Total NID Cost Allocated Aerial		= TotalGridNIDCost * PercentDropsAerial	
CS	TotalNIDCostAllocatedBuried	Total NID Cost Allocated Buried		= TotalGridNIDCost * (1 - PercentDropsAerial)	
CT	TotalDistributionCost	Total Distribution Cost		= TotalAerialCableDistributionCost + TotalBuriedCableDistributionCost + TotalUGCableDistributionCost + TotalPoleDistributionCost + TotalDistributionConduitCost + TotalGridNIDCost	
CU	AverageDistributionDistance	Average Distribution Distance		= 0.75 * AVERAGE (ULQBackboneandBranchCableLength, LLQBackboneBranchCableLength, URQBackboneBranchCableLength, LRQBackboneBranchCableLength) + AVERAGE (ULQDropLength, LLQDropLength, URQDropLength, LRQDropLength)	

Sheet: Subfeeder Part2

Col	Range Name	Column Name	Column Comment	Formula	Formula Comment
A	TerrainAdjustedPt2Distance	Terrain Adjusted Pt2 Distance		= SubfeederPart2Distance * SoilSlopeFactor	
B	MaxSizePt2FiberCables	# Max Size Pt2 Fiber Cables		= TRUNC (IF (NumberFibersPerGrid > MaxFiberSize, NumberFibersPerGrid / MaxFiberSize, 0))	
C	Pt2LastFiberCableSize	Last Pt2 Fiber Cable Size		= IF (NumberFibersPerGrid > 0, INDEX (FiberCableSize, MATCH ((NumberFibersPerGrid - (MaxFiberSize * MaxSizePt2FiberCables)) , FiberCableSize, - 1) , 1) , 0)	
D	Pt2CopperSize	Pt2 Size (If Copper)		= CHOOSE (GridPlantIndex, INDEX (FeederCableSize, MATCH (NetCopperPairs / FeederFill, FeederCableSize, - 1) , 1) , 0, 0)	
E	Pt2AerialCopperDistance	Pt2 Aerial Copper Distance		= IF (Pt2CopperSize = 0, 0, IF (GridPlantType = "Cable", VLOOKUP (Density, CHOOSE (CopperDepthCondition, CopperHardMixTable, CopperSoftMixTable, CopperNormMixTable) , 4) , 0) * TerrainAdjustedPt2Distance)	
F	SubfeederPt2AerialCopperMaterial	Subfeeder Pt2 Aerial Copper Material $		= IF (Pt2CopperSize = 0, 0, Pt2AerialCopperDistance * WaterCostMultiplier * CopperCostRatio * VLOOKUP (Pt2CopperSize, IF (CopperGauge = 26, AERCopper26Table, AERCopper24Table) , DensityColumnPointer, FALSE))	
G	Pt2AerialFiberDistance	Pt2 Aerial Fiber Distance		= IF (Pt2LastFiberCableSize = 0, 0, TerrainAdjustedPt2Distance * CHOOSE (GridPlantIndex, 0, 1, 1) * VLOOKUP (Density, CHOOSE (FiberDepthCondition, FbrLoopHardMixTable, FbrLoopSoftMixTable, FbrLoopNormMixTable) , 4))	
H	SubfeederPt2AerialFiberMaterial	Subfeeder Pt2 Aerial Fiber Material $		= IF (OR (Pt2AerialFiberDistance = 0, Pt2LastFiberCableSize = 0) , 0, Pt2AerialFiberDistance * (VLOOKUP (Pt2LastFiberCableSize, FiberAerialCostTable, DensityColumnPointer, FALSE) + (B3 * VLOOKUP (MaxFiberSize, FiberAerialCostTable, DensityColumnPointer)))) * FiberCostRatio	
I	Pt2BuriedCopperDistance	Pt2 Buried Copper Distance		= IF (Pt2CopperSize = 0, 0, TerrainAdjustedPt2Distance * CHOOSE (GridPlantIndex, 1, 0, 0) * VLOOKUP (Density, CHOOSE (CopperDepthCondition, CopperHardMixTable, CopperSoftMixTable, CopperNormMixTable) , 3))	
J	SubfeederPt2BuriedCopperMaterial	Subfeeder Pt2 Buried Copper Material $		= IF (Pt2CopperSize = 0, 0, Pt2BuriedCopperDistance * WaterCostMultiplier * CopperCostRatio * VLOOKUP (Pt2CopperSize, IF (CopperGauge = 26, BURCopper26Table, BURCopper24Table) , DensityColumnPointer, FALSE))	
K	Pt2BuriedFiberDistance	Pt2 Buried Fiber Distance		= IF (Pt2LastFiberCableSize = 0, 0, TerrainAdjustedPt2Distance * CHOOSE (GridPlantIndex, 0, 1, 1) * VLOOKUP (Density, CHOOSE (FiberDepthCondition, FbrLoopHardMixTable, FbrLoopSoftMixTable, FbrLoopNormMixTable) , 3))	
L	SubfeederPt2BuriedFiberMaterial	Subfeeder Pt2 Buried Fiber Material $		= IF (OR (Pt2BuriedFiberDistance = 0, Pt2LastFiberCableSize = 0) , 0, Pt2BuriedFiberDistance * (VLOOKUP (Pt2LastFiberCableSize, FiberBuriedCostTable, DensityColumnPointer, FALSE) + (MaxSizePt2FiberCables * VLOOKUP (MaxFiberSize, FiberBuriedCostTable, DensityColumnPointer)))) * FiberCostRatio	
M	Pt2UGCopperDistance	Pt2 Underground Copper Distance		= IF (Pt2CopperSize = 0, 0, TerrainAdjustedPt2Distance * CHOOSE (GridPlantIndex, 1, 0, 0) * VLOOKUP (Density, CHOOSE (CopperDepthCondition, CopperHardMixTable, CopperSoftMixTable, CopperNormMixTable) , 2))	
N	SubfeederPt2UGCopperMaterial	Subfeeder Pt2 UG Copper Material $		= IF (Pt2CopperSize = 0, 0, Pt2UGCopperDistance * WaterCostMultiplier * CopperCostRatio * VLOOKUP (Pt2CopperSize, IF (CopperGauge = 26, UGCopper26Table, UGCopper24Table) , DensityColumnPointer, FALSE))	
O	Pt2UGFiberDistance	Pt2 Underground Fiber Distance		= IF (Pt2LastFiberCableSize = 0, 0, TerrainAdjustedPt2Distance * CHOOSE (GridPlantIndex, 0, 1, 1) * VLOOKUP (Density, CHOOSE (FiberDepthCondition, FbrLoopHardMixTable, FbrLoopSoftMixTable, FbrLoopNormMixTable) , 2))	
P	SubfeederPt2UGFiberMaterial	Subfeeder Pt2 UG Fiber Material $		= IF (OR (Pt2UGFiberDistance = 0, Pt2LastFiberCableSize = 0) , 0, Pt2UGFiberDistance * (VLOOKUP (Pt2LastFiberCableSize, FiberUGCostTable, DensityColumnPointer, FALSE) + (MaxSizePt2FiberCables * VLOOKUP (MaxFiberSize, FiberUGCostTable, DensityColumnPointer)))) * FiberCostRatio	
Q	TotalPt2CopperMaterial	Total Pt2 Copper Material $		= SubfeederPt2AerialCopperMaterial + SubfeederPt2BuriedCopperMaterial + SubfeederPt2UGCopperMaterial	
R	TotalPt2FiberMaterial	Total Pt2 Fiber Material $		= SubfeederPt2AerialFiberMaterial + SubfeederPt2BuriedFiberMaterial + SubfeederPt2UGFiberMaterial	
S	NumberSubFdrPt2Manholes	Number Subfeeder Pt2 Manholes		= IF (MAX (Pt2UGCopperDistance, Pt2UGFiberDistance) < 1, 0, MAX (Pt2UGCopperDistance, Pt2UGFiberDistance) / VLOOKUP (Density, FeederSpacingTable, 2))	
T	NumberSubFdrPt2Ducts	Number Subfeeder Pt2 Ducts		= IF (Pt2UGFiberDistance + Pt2UGCopperDistance = 0, 0, MaxSizePt2FiberCables + 2)	
U	Pt2StandardManholeCost'	Pt2 Standard Manhole Cost		= IF (OR (NumberSubFdrPt2Manholes = 0, NumberSubFdrPt2Ducts > 9) , 0, VLOOKUP (NumberSubFdrPt2Ducts, CHOOSE (FiberDepthCondition, ManholeHardRockTable, ManholeSoftRockTable, ManholeNormalSurfaceTable) , DensityColumnPointer))	

V	Pt2ExpandedManholeCost	Pt2 Expanded Manhole Cost		= IF (OR (NumberSubFdrPt2Manholes = 0, NumberSubFdrPt2Ducts < = 9) , 0, VLOOKUP (9, CHOOSE (FiberDepthCondition, ManholeHardRockTable, ManholeSoftRockTable, ManholeNormalSurfaceTable) , DensityColumnPointer) + TRUNC (NumberSubFdrPt2Ducts / 9) * VLOOKUP (99, CHOOSE (FiberDepthCondition, ManholeHardRockTable, ManholeSoftRockTable, ManholeNormalSurfaceTable) , DensityColumnPointer))	
W	SubfeederPt2Aerial Structure	Subfeeder Pt2 Aerial Structure $		= IF ((Pt2AerialCopperDistance + Pt2AerialFiberDistance) < 1, 0, ((Pt2AerialCopperDistance + Pt2AerialFiberDistance) / VLOOKUP (Density, FeederSpacingTable, 3)) * WaterCostMultiplier * VLOOKUP (Density, CHOOSE (FiberDepthCondition, HardRockStructure, SoftRockStructure, NormalStructure) , 6))	
X	SubfeederPt2buriedcopperStructure	Subfeeder Pt2 Buried Copper Structure $		= IF (Pt2BuriedCopperDistance < 1, 0, WaterCostMultiplier * Pt2BuriedCopperDistance * VLOOKUP (Density, CHOOSE (CopperDepthCondition, HardRockStructure, SoftRockStructure, NormalStructure) , 4))	
Y	SubfeederPt2BuriedFiberStructure	Subfeeder Pt2 Buried Fiber Structure $		= IF (Pt2BuriedFiberDistance < 1, 0, WaterCostMultiplier * Pt2BuriedFiberDistance * VLOOKUP (Density, CHOOSE (FiberDepthCondition, HardRockStructure, SoftRockStructure, NormalStructure) , 4))	
Z	SubfeederPt2UGStructure	Subfeeder Pt2 Underground Structure $		= IF ((Pt2UGCopperDistance + Pt2UGFiberDistance) < 1, 0, WaterCostMultiplier * (Pt2UGCopperDistance + Pt2UGFiberDistance) * VLOOKUP (Density, CHOOSE (FiberDepthCondition, HardRockStructure, SoftRockStructure, NormalStructure) , 2) + NumberSubFdrPt2Ducts * VLOOKUP (0, ConduitDensityTable, DensityColumnPointer)) + NumberSubFdrPt2Manholes * IF (NumberSubFdrPt2Ducts > 9, Pt2ExpandedManholeCost, Pt2StandardManholeCost)	
AA	TotalPt2Structure	Total Pt2 Structure $		= SubfeederPt2AerialStructure + SubfeederPt2BuriedCopperStructure + SubfeederPt2BuriedFiberStructure + SubfeederPt2UGStructure	
AB	TotalSubfeederPart 2	Total Subfeeder Part 2 $		= TotalPt2CopperMaterial + TotalPt2FiberMaterial + TotalPt2Structure	

Sheet: SubFeeder

Col	Range Name	Column Name	Column Comment	Formula	Formula Comment
A	SubfeederNumber*	Subfeeder Number		= LEFT (FDICode, 5)	
B	SubfeederSegment Number	Subfeeder Segment Number		= IF (SubfeederNumber = PrevSubfeederNumber, PrevSubFeederSequenceNumber + 1, 1)	
C		Segment Type 1		= SegmentType1	
D	SubfeederSegment Type2	Subfeeder Segment Type 2		= IF (NextSubfeederSegmentNumber = 1, "None", IF (SegmentType1 = NextSegmentType1, NextSubfeederSegmentType2, NextSegmentType1))	
E	SubfeederCableTes tColumn	Copper Cable Runs through Subfeeder		= IF (OR (SegmentType1 = "Cable", SubfeederSegmentType2 = "Cable") , CLLIQuad, "None")	
F	SubfeederFiberTest Column	Fiber Cable Runs Through Subfeeder		= IF (OR (SegmentType1 = "Fiber", SubfeederSegmentType2 = "Fiber") , CLLIQuad, "None")	
G	PrevPrimarySubSe gmentDistance	Segment SubFeeder Distance		= IF (SubfeederSegmentNumber = 1, PrimarySubfeederDistance, PrimarySubfeederDistance - PrevPrimarySubfeederDistance)	
H	SlopeAdjSubfeeder Distance	Slope Adjusted Subfeeder Distance		= SegmentSubFeederdistance * SoilSlopeFactor	
I		Grid Plant Type		= GridPlantType	
J		Number Of Fibers From This Grid		= NumberFibersPerGrid	
K	CumulativeDLCS LinesperSubfeeder *	Cumulative DLC-S Lines Per Subfeeder		= CHOOSE (GridPlantIndex, 0, IF (OR (NextSubfeederSegmentNumber = 1, IsLastCLLIRecord) , TotalLinesServedInGrid / ElectronicFill, IF (AND (IsSameAsNextSubfeeder, IsSamePlantType) , (TotalLinesServedInGrid / ElectronicFill) + NextCumulativeDLCSLinesperSubfeeder, TotalLinesServedInGrid / ElectronicFill)) , 0)	
L	TotalDLCSLinesin SubFeeder	Total DLC-S Lines In Subfeeder		= IF (SubfeederSegmentNumber = 1, CumulativeDLCSLinesperSubfeeder, PrevTotalDLCSLinesinSubFeeder)	
M	NumberSubfeederF ibers*	Number Subfeeder Fibers This segment		= IF (IsSameAsNextSubfeeder, IF (AND (CumulativeDLCSLinesperSubfeeder > 0, CumulativeDLCSLinesperSubfeeder < 240, NumberFibersPerGrid = 4, NextNumberSubfeederFibers = 4) , 4, NumberFibersPerGrid + NextNumberSubfeederFibers) , NumberFibersPerGrid)	
N	IsDLCSFiberShare d	DLC-S System in Subfeeder		= IF (OR (NextSubfeederSegmentNumber = 1, LastDataRow) , IF (GridPlantType = "DLC - S", "Yes", "No") , IF (GridPlantType = "DLC - S", "Yes", NextIsDLCSFiberShared))	
O	NumberDLCSSyst emsServedinandbe yond	Number DLC-S Systems Served in Grid and beyond on Same SubFeeder		= IF (OR (NextSubfeederSegmentNumber = 1, IsLastCLLIRecord) , IF (GridPlantType = "DLC - S", 1, 0) , IF (GridPlantType = "DLC - S", NextNumberDLCSSytemsServedinandbeyond + 1, NextNumberDLCSSytemsServedinandbeyond))	
P	NumberDLCSSyst emsinSubFeeder	Number DLC-S Systems Served in SubFeeder		= IF (SubfeederSegmentNumber = 1, NumberDLCSSytemsServedinandbeyond, prevNumberDLCSSystemsinSubFeeder)	
Q	NumberDLCLFibe rsinandBeyond	Number of DLC-L Fibers in grid and Beyond on Same Subfeeder		= IF (OR (NextSubfeederSegmentNumber = 1, IsLastCLLIRecord) , IF (GridPlantType = "DLC - L", NumberFibersPerGrid, 0) , IF (GridPlantType = "DLC - L", NextNumberDLCLFibersinandBeyond + NumberFibersPerGrid, NextNumberDLCLFibersinandBeyond))	
R	NumberDLCSFibe rsinandBeyond	Number of DLC-S Fibers in grid and Beyond on Same Subfeeder		= NumberSubfeederFibers - NumberDLCLFibersinandBeyond	
S	NumberDLCSFibe rsInSubfeeder	Number DLC-S Fibers in Subfeeder		= IF (SubfeederSegmentNumber = 1, NumberDLCSFibersinandBeyond, PrevNumberDLCSFibersinSubfeeder)	
T	SubfeederFiberSize	Subfeeder Fiber Size		= IF (OR (PrimarySubfeederDistance = 0, SubfeederFiberTestColumn = "None") , 0, INDEX (FiberCableSize, MATCH (NumberSubfeederFibers, FiberCableSize, - 1) , 1))	

U	SubfeederCopperPairsthisGrid	Subfeeder Copper Pairs This Grid		= CHOOSE (GridPlantIndex, NetCopperPairs / FeederFill, 0, 0)	
V	CumulativeCopperSubfeederPairs*	Cumulative Copper Subfeeder Pairs		= IF (IsSameAsNextSubfeeder, SubfeederCopperPairsthisGrid + NextCumulativeCopperSubfeederPairs, SubfeederCopperPairsthisGrid)	
W	SubfeederMaxCopperCables	Number Of Max Size Subfeeder Copper Cables		= TRUNC (IF (CumulativeCopperSubfeederPairs > MaxFeederSize, CumulativeCopperSubfeederPairs / MaxFeederSize, 0))	
X	SubfeederCopperCableSize	Subfeeder Copper Cable Size		= IF (CumulativeCopperSubfeederPairs = 0, 0, INDEX (FeederCableSize, MATCH (CumulativeCopperSubfeederPairs - (MaxFeederSize * SubfeederMaxCopperCables) , FeederCableSize, - 1) , 1))	
Y	SubfeederUGDistance	Subfeeder Underground Distance		= IF (SubfeederFiberTestColumn = "None", VLOOKUP (Density, CopperNormMixTable, 2) , VLOOKUP (Density, FbrLoopNormMixTable, 2)) * SlopeAdjSubfeederDistance * SoilSlopeFactor	
Z	SubfeederBURDistance	Subfeeder Buried Distance		= IF (SubfeederFiberTestColumn = "None", VLOOKUP (Density, CopperNormMixTable, 3) , VLOOKUP (Density, FbrLoopNormMixTable, 3)) * SlopeAdjSubfeederDistance * SoilSlopeFactor	
AA	SubfeederAERDistance	Subfeeder Aerial Distance		= IF (SubfeederFiberTestColumn = "None", VLOOKUP (Density, CopperNormMixTable, 4) , VLOOKUP (Density, FbrLoopNormMixTable, 4)) * SlopeAdjSubfeederDistance * SoilSlopeFactor	
AB	SubfeederUGCopperCable	Subfeeder Underground Copper Cable $		= IF (OR (SubfeederCableTestColumn = "None", SubfeederUGDistance = 0) , 0, SubfeederUGDistance * WaterCostMultiplier * CopperCostRatio * (VLOOKUP (SubfeederCopperCableSize, IF (CopperGauge = 26, UGCopper26Table, UGCopper24Table) , DensityColumnPointer, FALSE)) + (SubfeederMaxCopperCables * VLOOKUP (MaxFeederSize, IF (CopperGauge = 26, UGCopper26Table, UGCopper24Table) , DensityColumnPointer, FALSE)))	
AC	SubfeederBuriedCopperCable	Subfeeder Buried Copper Cable $		= IF (OR (SubfeederBURDistance = 0, SubfeederCableTestColumn = "None") , 0, SubfeederBURDistance * WaterCostMultiplier * CopperCostRatio * (VLOOKUP (SubfeederCopperCableSize, IF (CopperGauge = 26, BURCopper26Table, BURCopper24Table) , DensityColumnPointer, FALSE)) + (SubfeederMaxCopperCables * VLOOKUP (MaxFeederSize, IF (CopperGauge = 26, BURCopper26Table, BURCopper24Table) , DensityColumnPointer, FALSE)))	
AD	SubfeederAerialCopperCable	Subfeeder Aerial Copper Cable $		= IF (OR (SubfeederAERDistance = 0, SubfeederCableTestColumn = "None") , 0, SubfeederAERDistance * WaterCostMultiplier * CopperCostRatio * (VLOOKUP (SubfeederCopperCableSize, IF (CopperGauge = 26, AERCopper26Table, AERCopper24Table) , DensityColumnPointer, FALSE)) + (SubfeederMaxCopperCables * VLOOKUP (MaxFeederSize, IF (CopperGauge = 26, AERCopper26Table, AERCopper24Table) , DensityColumnPointer, FALSE)))	
AE	CopperSubfeederTotalMaterial	Copper Subfeeder Total Material $		= SubfeederUGCopperCable + SubfeederBuriedCopperCable + SubfeederAerialCopperCable	
AF	UGFiberSubfeederMaterial	Underground Fiber Subfeeder Material $		= IF (OR (SubfeederFiberTestColumn = "None", SubfeederUGDistance = 0) , 0, SubfeederUGDistance * (VLOOKUP (SubfeederFiberSize, FiberUGCostTable, DensityColumnPointer, FALSE) * FiberCostRatio))	
AG	BuriedFiberSubfeederMaterial	Buried Fiber Subfeeder Material $		= IF (OR (SubfeederFiberTestColumn = "None", SubfeederBURDistance = 0) , 0, SubfeederBURDistance * (VLOOKUP (SubfeederFiberSize, FiberBuriedCostTable, DensityColumnPointer, FALSE) * FiberCostRatio))	
AH	AerialFiberSubfeederMaterial	Aerial Fiber Subfeeder Material $		= IF (OR (SubfeederFiberTestColumn = "None", SubfeederAERDistance = 0) , 0, SubfeederAERDistance * (VLOOKUP (SubfeederFiberSize, FiberAerialCostTable, DensityColumnPointer, FALSE) * FiberCostRatio))	
AI	FiberSubfeederTotalMaterial	Fiber Subfeeder Total Material $		= UGFiberSubfeederMaterial + BuriedFiberSubfeederMaterial + AerialFiberSubfeederMaterial	
AJ	CopperSubfeederStructureRatio	Copper Subfeeder Structure Ratio		= IF (CopperSubfeederTotalMaterial = 0, 0, IF (SubfeederMaxCopperCables > 0, Over4200, IF (AND (CopperSubfeederTotalMaterial > 0, FiberSubfeederTotalMaterial > 0) , VLOOKUP (SubfeederCopperCableSize, StructureAllocationTable, 2) , 1)))	
AK	FiberSubfeederStructureRatio	Fiber Subfeeder Structure Ratio		= 1 - CopperSubfeederStructureRatio	
AL	NumberSubfeederDucts	Number Subfeeder Ducts		= IF ((SubfeederUGCopperCable + UGFiberSubfeederMaterial) < 1, 0, 1 + IF (SubfeederUGCopperCable > 0, SubfeederMaxCopperCables + 1, 0) + IF (UGFiberSubfeederMaterial > 0, 1, 0))	

AM	NumberSubfeeder Manholes	Number Subfeeder Manholes		= IF (NumberSubfeederDucts < 1, 0, CEILING (SubfeederUGDistance / VLOOKUP (Density, FeederSpacingTable, 2) , 1))	
AN	SubfeederStandard ManholeCost	Subfeeder Standard Manhole Cost		= IF (OR (NumberSubfeederManholes = 0, NumberSubfeederDucts > 9) , 0, VLOOKUP (NumberSubfeederDucts, CHOOSE (FiberDepthCondition, ManholeHardRockTable, ManholeSoftRockTable, ManholeNormalSurfaceTable) , DensityColumnPointer))	
AO	SubfeederExpanded ManholeCost	Subfeeder Expanded Manhole Cost		= IF (OR (NumberSubfeederManholes = 0, NumberSubfeederDucts < = 9) , 0, VLOOKUP (9, CHOOSE (FiberDepthCondition, ManholeHardRockTable, ManholeSoftRockTable, ManholeNormalSurfaceTable) , DensityColumnPointer) + TRUNC (NumberSubfeederDucts / 9) * VLOOKUP (99, CHOOSE (FiberDepthCondition, ManholeHardRockTable, ManholeSoftRockTable, ManholeNormalSurfaceTable) , DensityColumnPointer))	
AP	NumberSubfeederPoles	Number Subfeeder Poles		= IF (SubfeederAERDistance < 1, 0, CEILING (SubfeederAERDistance / VLOOKUP (Density, FeederSpacingTable, 3) , 1) + IF (SubfeederSegmentNumber = 1, 1, 0))	
AQ	CopperSubfeederUGStructure	Copper Subfeeder Underground Structure $		= IF (SubfeederUGDistance = 0, 0, CopperSubfeederStructureRatio * WaterCostMultiplier * (IF (NumberSubfeederDucts < = 9, SubfeederStandardManholeCost, SubfeederExpandedManholeCost) * NumberSubfeederManholes + SubfeederUGDistance * (NumberSubfeederDucts * VLOOKUP (0, ConduitDensityTable, DensityColumnPointer) + VLOOKUP (Density, CHOOSE (FiberDepthCondition, HardRockStructure, SoftRockStructure, NormalStructure) , 2))))	
AR	CopperSubfeederBuriedStructure	Copper Subfeeder Buried Structure $		= IF (SubfeederBURDistance = 0, 0, CopperSubfeederStructureRatio * WaterCostMultiplier * SubfeederBURDistance * IF (SubfeederBURDistance = 0, VLOOKUP (Density, CHOOSE (CopperDepthCondition, HardRockStructure, SoftRockStructure, NormalStructure) , 4) , VLOOKUP (Density, CHOOSE (FiberDepthCondition, HardRockStructure, SoftRockStructure, NormalStructure) , 4)))	
AS	CopperSubfeederAerialStructure	Copper Subfeeder Aerial Structure $		= IF (NumberSubfeederPoles = 0, 0, CopperSubfeederStructureRatio * WaterCostMultiplier * NumberSubfeederPoles * VLOOKUP (Density, CHOOSE (FiberDepthCondition, HardRockStructure, SoftRockStructure, NormalStructure) , 6))	
AT	CopperSubfeederTotalStructure	Copper Subfeeder Total Structure $		= CopperSubfeederAerialStructure + CopperSubfeederBuriedStructure + CopperSubfeederUGStructure	
AU	FiberSubfeederUG Structure	Fiber Subfeeder Underground Structure $		= IF (SubfeederUGDistance = 0, 0, FiberSubfeederStructureRatio * WaterCostMultiplier * (IF (NumberSubfeederDucts < = 9, SubfeederStandardManholeCost, SubfeederExpandedManholeCost) * NumberSubfeederManholes + SubfeederUGDistance * (NumberSubfeederDucts * VLOOKUP (0, ConduitDensityTable, DensityColumnPointer) + VLOOKUP (Density, CHOOSE (FiberDepthCondition, HardRockStructure, SoftRockStructure, NormalStructure) , 2))))	
AV	BuriedFiberSubfederStructure	Buried Fiber Subfeeder Structure $		= IF (SubfeederBURDistance = 0, 0, FiberSubfeederStructureRatio * WaterCostMultiplier * SubfeederBURDistance * IF (SubfeederBURDistance = 0, VLOOKUP (Density, CHOOSE (CopperDepthCondition, HardRockStructure, SoftRockStructure, NormalStructure) , 4) , VLOOKUP (Density, CHOOSE (FiberDepthCondition, HardRockStructure, SoftRockStructure, NormalStructure) , 4)))	
AW	AerialFiberSubfederStructure	Aerial Fiber Subfeeder Structure $		= IF (NumberSubfeederPoles = 0, 0, FiberSubfeederStructureRatio * WaterCostMultiplier * NumberSubfeederPoles * VLOOKUP (Density, CHOOSE (FiberDepthCondition, HardRockStructure, SoftRockStructure, NormalStructure) , 6))	
AX	FiberSubfeederTotalStructure	Fiber Subfeeder Total Structure $		= FiberSubfeederUGStructure + BuriedFiberSubfeederStructure + AerialFiberSubfeederStructure	
AY	TotalCopperSubfeeder	Total Copper Subfeeder $		= CopperSubfeederTotalMaterial + CopperSubfeederTotalStructure	
AZ	TotalCopperSubfederCostPerPair	Total Copper Subfeeder Cost Per Pair		= IF (CumulativeCopperSubfeederPairs = 0, 0, TotalCopperSubfeeder / CumulativeCopperSubfeederPairs)	
BA	CumulativeCopper SubfeederCostperPair*	Cumulative Copper Subfeeder Cost Per Pair		= IF (SegmentType1 = "Fiber", 0, IF (IsFirstSubfeederSegment, TotalCopperSubfeederCostPerPair, TotalCopperSubfeederCostPerPair + PrevCumulativeCopperSubfeederCostperPair))	
BB	TotalFiberSubfeeder	Total Fiber Subfeeder $		= FiberSubfeederTotalMaterial + FiberSubfeederTotalStructure	
BC	TotalFiberSubfeederCostPerPair	Total Fiber Subfeeder Cost Per Fiber		= IF (NumberSubfeederFibers = 0, 0, TotalFiberSubfeeder / NumberSubfeederFibers)	
BD	CumulativeFiberSubfeederCostperFiber*	Cumulative Fiber Subfeeder Cost Per Fiber		= IF (IsFirstSubfeederSegment, TotalFiberSubfeederCostPerPair, TotalFiberSubfeederCostPerPair + PrevCumulativeFiberSubfeederCostperFiber)	

BE	CumSubFdrUGCopperCostperLine	Cumulative Sub Feeder Underground Copper Cost per Line		= IF (CumulativeCopperSubfeederPairs = 0, 0, IF (IsFirstSubfeederSegment, SubfeederUGCopperCable / CumulativeCopperSubfeederPairs, SubfeederUGCopperCable / CumulativeCopperSubfeederPairs + PrevCumSubFdrUGCopperCostperLine))	
BF	CumSubFdrBurCopperCostperLine	Cumulative Sub Feeder Buried Copper Cost per Line		= IF (CumulativeCopperSubfeederPairs = 0, 0, IF (IsFirstSubfeederSegment, SubfeederBuriedCopperCable / CumulativeCopperSubfeederPairs, SubfeederBuriedCopperCable / CumulativeCopperSubfeederPairs + PrevCumSubFdrBURCopperCostperLine))	
BG	CumSubFdrAERCopperCostperLine	Cumulative Sub Feeder Aerial Copper Cost per Line		= IF (CumulativeCopperSubfeederPairs = 0, 0, IF (IsFirstSubfeederSegment, SubfeederAerialCopperCable / CumulativeCopperSubfeederPairs, SubfeederAerialCopperCable / CumulativeCopperSubfeederPairs + PrevCumSubFdrAERCopperCostperLine))	
BH	CumSubFdrUGCopperStructureCostperLine	Cumulative Sub Feeder Underground Copper Strucutre Cost per Line		= IF (CumulativeCopperSubfeederPairs = 0, 0, IF (IsFirstSubfeederSegment, CopperSubfeederUGStructure / CumulativeCopperSubfeederPairs, CopperSubfeederUGStructure / CumulativeCopperSubfeederPairs + PrevCumSubFdrUGCopperStructureCostperLine))	
BI	CumSubFdrBurCopperStructureCostperLine	Cumulative Sub Feeder Buried Copper Structure Cost per Line		= IF (CumulativeCopperSubfeederPairs = 0, 0, IF (IsFirstSubfeederSegment, CopperSubfeederBuriedStructure / CumulativeCopperSubfeederPairs, CopperSubfeederBuriedStructure / CumulativeCopperSubfeederPairs + PrevCumSubFdrBURCopperStructureCostperLine))	
BJ	CumSubFdrAerCopperStructureCostperLine	Cumulative Sub Feeder Aerial Copper Structure Cost per Line		= IF (CumulativeCopperSubfeederPairs = 0, 0, IF (IsFirstSubfeederSegment, CopperSubfeederAerialStructure / CumulativeCopperSubfeederPairs, CopperSubfeederAerialStructure / CumulativeCopperSubfeederPairs + PrevCumSubFdrAERCopperStructureCostperLine))	
BK	CumSubFdrUGFiberCostperFiber	Cumulative Sub Feeder Underground Fiber Cost per Fiber		= IF (NumberSubfeederFibers = 0, 0, IF (IsFirstSubfeederSegment, UGFiberSubfeederMaterial / NumberSubfeederFibers, UGFiberSubfeederMaterial / NumberSubfeederFibers + PrevCumSubFdrUGFiberCostperFiber))	
BL	CumSubFdrBurFiberCostperFiber	Cumulative Sub Feeder Buried Fiber Cost per Fiber		= IF (NumberSubfeederFibers = 0, 0, IF (IsFirstSubfeederSegment, BuriedFiberSubfeederMaterial / NumberSubfeederFibers, BuriedFiberSubfeederMaterial / NumberSubfeederFibers + PrevCumSubFdrBURFiberCostperFiber))	
BM	CumSubFdrAERFiberCostperFiber	Cumulative Sub Feeder Aerial Fiber Cost per Fiber		= IF (NumberSubfeederFibers = 0, 0, IF (IsFirstSubfeederSegment, AerialFiberSubfeederMaterial / NumberSubfeederFibers, AerialFiberSubfeederMaterial / NumberSubfeederFibers + PrevCumSubFdrAERFiberCostperFiber))	
BN	CumSubFdrUGFiberStructureCostperFiber	Cumulative Sub Feeder Underground Fiber Strucutre Cost per Fiber		= IF (NumberSubfeederFibers = 0, 0, IF (IsFirstSubfeederSegment, FiberSubfeederUGStructure / NumberSubfeederFibers, FiberSubfeederUGStructure / NumberSubfeederFibers + PrevCumSubFdrUGFiberStrucutreCostperFiber))	
BO	CumSubFdrBurFiberStructureCostperFiber	Cumulative Sub Feeder Buried Fiber Structure Cost per Fiber		= IF (NumberSubfeederFibers = 0, 0, IF (IsFirstSubfeederSegment, BuriedFiberSubfeederStructure / NumberSubfeederFibers, BuriedFiberSubfeederStructure / NumberSubfeederFibers + PrevCumSubFdrBURFiberStrucutreCostperFiber))	
BP	CumSubFdrAerFiberStructureCostperFiber	Cumulative Sub Feeder Aerial Fiber Structure Cost per Fiber		= IF (NumberSubfeederFibers = 0, 0, IF (IsFirstSubfeederSegment, AerialFiberSubfeederStructure / NumberSubfeederFibers, AerialFiberSubfeederStructure / NumberSubfeederFibers + PrevCumSubFdrAERFiberStrucutreCostperFiber))	

BQ	CumDLCSSubFdr UGFiberCostperLi ne	Cumulative DLC-S Sub Feeder Underground Fiber Cost per Line		= IF (CumulativeDLCSLinesperSubfeeder = 0, 0, IF (IsFirstSubfeederSegment, IF (IsDLCSFiberShared = "No", 0, (NumberDLCSFibersinandBeyond / NumberSubfeederFibers) * (UGFiberSubfeederMaterial / CumulativeDLCSLinesperSubfeeder)) , IF (IsDLCSFiberShared = "No", 0, (NumberDLCSFibersinandBeyond / NumberSubfeederFibers) * (UGFiberSubfeederMaterial / CumulativeDLCSLinesperSubfeeder)) + PrevCumDLCSSubFdrUGFiberCostperLine))	
BR	CumDLCSSubFdr BurFiberCostperLi ne	Cumulative DLC-S Sub Feeder Buried Fiber Cost per Line		= IF (CumulativeDLCSLinesperSubfeeder = 0, 0, IF (IsFirstSubfeederSegment, IF (IsDLCSFiberShared = "No", 0, (NumberDLCSFibersinandBeyond / NumberSubfeederFibers) * (BuriedFiberSubfeederMaterial / CumulativeDLCSLinesperSubfeeder)) , IF (IsDLCSFiberShared = "No", 0, (NumberDLCSFibersinandBeyond / NumberSubfeederFibers) * (BuriedFiberSubfeederMaterial / CumulativeDLCSLinesperSubfeeder)) + PrevCumDLCSSubFdrBURFiberCostperLine))	
BS	CumDLCSSubFdr AERFiberCostperL ine	Cumulative DLC-S Sub Feeder Aerial Fiber Cost per Line		= IF (CumulativeDLCSLinesperSubfeeder = 0, 0, IF (IsFirstSubfeederSegment, IF (IsDLCSFiberShared = "No", 0, (NumberDLCSFibersinandBeyond / NumberSubfeederFibers) * (AerialFiberSubfeederMaterial / CumulativeDLCSLinesperSubfeeder)) , IF (IsDLCSFiberShared = "No", 0, (NumberDLCSFibersinandBeyond / NumberSubfeederFibers) * (AerialFiberSubfeederMaterial / CumulativeDLCSLinesperSubfeeder)) + PrevCumDLCSSubFdrAERFiberCostperLine))	
BT	CumDLCSSubFdr UGFiberStructure CostperLine	Cumulative DLC-S Sub Feeder Underground Fiber Strucutre Cost per Line		= IF (CumulativeDLCSLinesperSubfeeder = 0, 0, IF (IsFirstSubfeederSegment, IF (IsDLCSFiberShared = "No", 0, (NumberDLCSFibersinandBeyond / NumberSubfeederFibers) * (FiberSubfeederUGStructure / CumulativeDLCSLinesperSubfeeder)) , IF (IsDLCSFiberShared = "No", 0, (NumberDLCSFibersinandBeyond / NumberSubfeederFibers) * (FiberSubfeederUGStructure / CumulativeDLCSLinesperSubfeeder)) + PrevCumDLCSSubFdrUGFiberStructureCostperLine))	
BU	CumDLCSSubFdr BurFiberStructure CostperLine	Cumulative DLC-S Sub Feeder Buried Fiber Structure Cost per Line		= IF (CumulativeDLCSLinesperSubfeeder = 0, 0, IF (IsFirstSubfeederSegment, IF (IsDLCSFiberShared = "No", 0, (NumberDLCSFibersinandBeyond / NumberSubfeederFibers) * (BuriedFiberSubfeederStructure / CumulativeDLCSLinesperSubfeeder)) , IF (IsDLCSFiberShared = "No", 0, (NumberDLCSFibersinandBeyond / NumberSubfeederFibers) * (BuriedFiberSubfeederStructure / CumulativeDLCSLinesperSubfeeder)) + PrevCumDLCSSubFdrBURFiberStrucutreCostperLine))	
BV	CumDLCSSubFdr AerFiberStructure CostperLine	Cumulative DLC-S Sub Feeder Aerial Fiber Structure Cost per Line		= IF (CumulativeDLCSLinesperSubfeeder = 0, 0, IF (IsFirstSubfeederSegment, IF (IsDLCSFiberShared = "No", 0, (NumberDLCSFibersinandBeyond / NumberSubfeederFibers) * (AerialFiberSubfeederStructure / CumulativeDLCSLinesperSubfeeder)) , IF (IsDLCSFiberShared = "No", 0, (NumberDLCSFibersinandBeyond / NumberSubfeederFibers) * (AerialFiberSubfeederStructure / CumulativeDLCSLinesperSubfeeder)) + PrevCumDLCSSubFdrAERFiberStrucutreCostperLine))	
BW	IsFirstSubfeederSe gement	First Subfeeder Segment		= SubfeederSegmentNumber = 1	
BX	IsLastCLLIRecord	Last CLLI Record		= IsLastCLLIRecord	
BY	IsSameAsNextQua drant	Same As Next Quadrant		= IsSameAsNextCLLIQuad	
BZ	IsSameAsNextSubf eeder	Same As Next Subfeeder		= (SubfeederNumber = NextSubfeederNumber)	

Sheet: Main Feeder

Col	Range Name	Column Name	Column Comment	Formula	Formula Comment
A		Quadrant Number		= CLLIQuad	
B		Main Feeder Branch Number		= MainFeederBranch	
C		Feeder Sequence Number	Data Source: Grid Demographics	= MainFeederSequenceNumber	
D		Feeder Segment Sequence Number		= FeederSegmentSequenceNumber	
E	MainFdrSegmentDistance	Main Feeder Segment Distance		= IF (IsFirstSegmentInBranch, MainFeederDistance, IF (IsSameAsPrevFeederBranch, MainFeederDistance - PrevMainFeederDistance, MainFeederDistance - 10000))	
F		Segment Type 1		= SegmentType1	
G		Segment Type 2		= SegmentType2	
H	CableTestColumn	Copper Cable Runs through Grid		= IF (OR (SegmentType1 = "Cable", SegmentType2 = "Cable") , CLLIQuad, "None")	
I	CableBranchTestColumn	Same Branch And Cable		= IF (SegmentType1 = "Cable", IF (MainFeederBranch = "0", "", IF (MainFeederBranch = "1", CLLIQuad&"2", CLLIQuad&"1")) , "")	
J	FiberTestColumn	Fiber Cable Runs Through Grid		= IF (OR (SegmentType1 = "Fiber", SegmentType2 = "Fiber") , CLLIQuad, "None")	
K	FiberBranchTestColumn	Same Branch And Fiber		= IF (QuadrantContainFiber, IF (MainFeederBranch = "0", "", IF (MainFeederBranch = "1", CLLIQuad&"2", CLLIQuad&"1")) , "")	
L	WorkingPrsfromSubfeeders	Working Pairs from Subfeeders		= IF (IsFirstSubfeederSegment, IF (PrimarySubfeederDistance = 0, NetCopperPairs, CumulativeCopperSubfeederPairs * FeederFill) , 0)	
M	MainFdrCopperWrkgLines	Main Feeder Copper Working Lines		= SUMIF (OFFSET (CableTestColumn, 0, 0, LastDataRow, 1) , CLLIQuad, OFFSET ($L3, 0, 0, LastDataRow, 1)) - SUMIF (OFFSET (CableBranchTestColumn, 0, 0, LastDataRow, 1) , CLLISegment, OFFSET ($L3, 0, 0, LastDataRow, 1))	
N	WorkingFibersfromSubfeeders	Working Fibers From Subfeeders		= IF (IsFirstSubfeederSegment, IF (PrimarySubfeederDistance = 0, NumberFibersPerGrid, NumberSubfeederFibers) , 0)	
O	MainFdrWrkgfibers	Main Feeder Working fibers		= SUMIF (OFFSET (FiberTestColumn, 0, 0, LastDataRow, 1) , CLLIQuad, OFFSET ($N3, 0, 0, LastDataRow, 1)) - SUMIF (OFFSET (FiberBranchTestColumn, 0, 0, LastDataRow, 1) , CLLISegment, OFFSET ($N3, 0, 0, LastDataRow, 1))	
P	MainFdrFibersNeeded	Main Feeder Fibers Needed		= MainFdrWrkgFibers / FeederFill	
Q	CopperMainFdrPrsNeeded	Copper Main Feeder Pairs Needed		= MainFdrCopperWrkgLines / FeederFill	
R	NumberMaxSizeMainFdrCopperCables	Number of Max Size Main Feeder Copper Cables		= TRUNC (IF (CopperMainFdrPrsNeeded > MaxFeederSize, CopperMainFdrPrsNeeded / MaxFeederSize, 0))	
S	ResidualCopperMainFdrSize	Residual Copper Main Feeder Size		= IF (CopperMainFdrPrsNeeded > 0, INDEX (FeederCableSize, MATCH (CopperMainFdrPrsNeeded - (MaxFeederSize * NumberMaxSizeMainFdrCopperCables) , FeederCableSize, - 1) , 1) , 0)	
T	CopperMainFdrStructureRatio	Copper Main Feeder Structure %		= IF (CopperMainFdrPrsNeeded < 1, 0, IF (NumberMaxSizeMainFdrCopperCables > 0, Over4200, IF (MainFdrFibersNeeded > 0, VLOOKUP (ResidualCopperMainFdrSize, StructureAllocationTable, 2) , 1)))	
U	FiberMainFdrStructureRatio	Fiber Main Feeder Structure %		= 1 - CopperMainFdrStructureRatio	
V	NumberMaxSizeMainFdrFiberCables	Number Max Size Main Feeder Fiber Cables		= TRUNC (IF (MainFdrFibersNeeded > MaxFiberSize, MainFdrFibersNeeded / MaxFiberSize, 0))	
W	ResidualFiberMainFdrSize	Residual Fiber Main Feeder Size		= IF (MainFdrFibersNeeded = 0, 0, INDEX (FiberCableSize, MATCH (MainFdrFibersNeeded - (MaxFiberSize * NumberMaxSizeMainFdrFiberCables) , FiberCableSize, - 1) , 1))	

X	MainFdrSegmentUGDistance	Main Feeder Underground Distance		= IF (FiberTestColumn = "None", VLOOKUP (Density, CopperNormMixTable, 2) , VLOOKUP (Density, FbrLoopNormMixTable, 2)) * MainFdrSegmentDistance * SoilSlopeFactor	
Y	MainFdrSegmentBurDistance	Main Feeder Buried Distance		= IF (FiberTestColumn = "None", VLOOKUP (Density, CopperNormMixTable, 3) , VLOOKUP (Density, FbrLoopNormMixTable, 3)) * MainFdrSegmentDistance * SoilSlopeFactor	
Z	MainFdrSegmentAerDistance	Main Feeder Aerial Distance		= IF (FiberTestColumn = "None", VLOOKUP (Density, CopperNormMixTable, 4) , VLOOKUP (Density, FbrLoopNormMixTable, 4)) * MainFdrSegmentDistance * SoilSlopeFactor	
AA	MainFdrSegmentUGCopperMaterial	Main Feeder Segment Underground Copper Material $		= IF (OR (CableTestColumn = "None", MainFdrSegmentUGDistance = 0) , 0, MainFdrSegmentUGDistance * (IF (ResidualCopperMainFdrSize = 0, 0, VLOOKUP (ResidualCopperMainFdrSize, IF (CopperGauge = 26, UGCopper26Table, UGCopper24Table) , DensityColumnPointer, FALSE)) + (NumberMaxSizeMainFdrCopperCables * VLOOKUP (MaxFeederSize, IF (CopperGauge = 26, UGCopper26Table, UGCopper24Table) , DensityColumnPointer, FALSE)))) * CopperCostRatio	
AB	MainFdrSegmentBurCopperMaterial	Main Feeder Segment Buried Copper Material $		= IF (OR (CableTestColumn = "None", 'MainFdrSegmentBurDistance' = 0) , 0, 'MainFdrSegmentBurDistance' * (IF (ResidualCopperMainFdrSize = 0, 0, VLOOKUP (ResidualCopperMainFdrSize, IF (CopperGauge = 26, BURCopper26Table, BURCopper24Table) , DensityColumnPointer, FALSE)) + (NumberMaxSizeMainFdrCopperCables * VLOOKUP (MaxFeederSize, IF (CopperGauge = 26, BURCopper26Table, BURCopper24Table) , DensityColumnPointer, FALSE)))) * CopperCostRatio	
AC	MainFdrSegmentAerCopperMaterial	Main Feeder Segment Aerial Copper Material $		= IF (OR (CableTestColumn = "None", MainFdrSegmentAerDistance = 0) , 0, MainFdrSegmentAerDistance * (IF (ResidualCopperMainFdrSize = 0, 0, VLOOKUP (ResidualCopperMainFdrSize, IF (CopperGauge = 26, AERCopper26Table, AERCopper24Table) , DensityColumnPointer, FALSE)) + (NumberMaxSizeMainFdrCopperCables * VLOOKUP (MaxFeederSize, IF (CopperGauge = 26, AERCopper26Table, AERCopper24Table) , DensityColumnPointer, FALSE)))) * CopperCostRatio	
AD	MainFdrSegmentAerCopperMaterial	Total Main Feeder Copper Material $		= MainFdrSegmentUGCopperMaterial + MainFdrSegmentBurCopperMaterial + MainFdrSegmentAerCopperMaterial	
AE	MainFdrSegmentUGFiberMaterial	Main Feeder Segment Underground Fiber Material $		= IF (OR (FiberTestColumn = "None", MainFdrSegmentUGDistance = 0) , 0, MainFdrSegmentUGDistance * (IF (ResidualFiberMainFdrSize = 0, 0, VLOOKUP (ResidualFiberMainFdrSize, FiberUGCostTable, DensityColumnPointer, FALSE)) + (NumberMaxSizeMainFdrFiberCables * VLOOKUP (MaxFiberSize, FiberUGCostTable, DensityColumnPointer)))) * FiberCostRatio	
AF	MainFdrSegmentBurFiberMaterial	Main Feeder Segment Buried Fiber Material $		= IF (OR (FiberTestColumn = "None", 'MainFdrSegmentBurDistance' = 0) , 0, 'MainFdrSegmentBurDistance' * (IF (ResidualFiberMainFdrSize = 0, 0, VLOOKUP (ResidualFiberMainFdrSize, FiberBuriedCostTable, DensityColumnPointer, FALSE)) + (NumberMaxSizeMainFdrFiberCables * VLOOKUP (MaxFiberSize, FiberBuriedCostTable, DensityColumnPointer)))) * FiberCostRatio	
AG	MainFdrSegmentAerFiberMaterial	Main Feeder Segment Aerial Fiber Material $		= IF (OR (FiberTestColumn = "None", MainFdrSegmentAerDistance = 0) , 0, MainFdrSegmentAerDistance * (IF (ResidualFiberMainFdrSize = 0, 0, VLOOKUP (ResidualFiberMainFdrSize, FiberAerialCostTable, DensityColumnPointer, FALSE)) + (NumberMaxSizeMainFdrFiberCables * VLOOKUP (MaxFiberSize, FiberAerialCostTable, DensityColumnPointer)))) * FiberCostRatio	
AH	TotalMainFdrFiberMaterial	Total Main Feeder Fiber Material $		= MainFdrSegmentUGFiberMaterial + MainFdrSegmentBurFiberMaterial + MainFdrSegmentAerFiberMaterial	
AI	NumberMainFeederDucts	Number Main Feeder Ducts		= IF (MainFdrSegmentUGDistance < 1, 0, IF (ResidualCopperMainFdrSize > 0, 1, 0) + NumberMaxSizeMainFdrCopperCables + IF (ResidualFiberMainFdrSize > 0, 1, 0) + IF (AirlineDistance = "", 0, 1))	
AJ	NumberMainFdrManholes	Number Main Feeder Manholes		= IF (NumberMainFeederDucts < 1, 0, CEILING (MainFdrSegmentUGDistance / VLOOKUP (Density, FeederSpacingTable, 2) , 1) + IF (AirlineDistance = "", 0, IF (FeederSegmentSequenceNumber = 1, 1, 0)))	
AK	MainFeederStandardManholeCost	Main Feeder Standard Manhole Cost		= IF (NumberMainFdrManholes = 0, 0, IF (NumberMainFeederDucts < = 9, VLOOKUP (NumberMainFeederDucts, CHOOSE (FiberDepthCondition, ManholeHardRockTable, ManholeSoftRockTable, ManholeNormalSurfaceTable) , DensityColumnPointer) , 0))	
AL	MainFeederExpandedManholeCost	Main Feeder Expanded Manhole Cost		= IF (NumberMainFeederDucts = 0, 0, IF (NumberMainFeederDucts < = 9, 0, VLOOKUP (9, CHOOSE (FiberDepthCondition, ManholeHardRockTable, ManholeSoftRockTable, ManholeNormalSurfaceTable) , DensityColumnPointer)) + TRUNC (NumberMainFeederDucts / 9) * VLOOKUP (99, CHOOSE (FiberDepthCondition, ManholeHardRockTable, ManholeSoftRockTable, ManholeNormalSurfaceTable) , DensityColumnPointer))	
AM	NumberMainFdrPoles	Number Main Feeder Poles		= IF (MainFdrSegmentAerDistance < 1, 0, (CEILING (MainFdrSegmentAerDistance / VLOOKUP (Density, FeederSpacingTable, 3) , 1)) + IF (AirlineDistance = "", 0, IF (FeederSegmentSequenceNumber = 1, 1, 0)))	

AN	MainFeederUGCopperStructure	Main Feeder Underground Copper Structure $		= IF (MainFdrSegmentUGDistance = 0, 0, CopperMainFdrStructureRatio * WaterCostMultiplier * (IF (NumberMainFeederDucts < = 9, MainFeederStandardManholeCost, MainFeederExpandedManholeCost) * NumberMainFdrManholes + MainFdrSegmentUGDistance * (NumberMainFeederDucts * VLOOKUP (0, ConduitDensityTable, DensityColumnPointer) + VLOOKUP (Density, CHOOSE (FiberDepthCondition, HardRockStructure, SoftRockStructure, NormalStructure) , 2))))	
AO	MainFeederBuriedCopperStructure	Main Feeder Buried Copper Structure $		= IF ('MainFdrSegmentBurDistance' = 0, 0, CopperMainFdrStructureRatio * WaterCostMultiplier * 'MainFdrSegmentBurDistance' * IF (FiberTestColumn = "None", VLOOKUP (Density, CHOOSE (CopperDepthCondition, HardRockStructure, SoftRockStructure, NormalStructure) , 4) , VLOOKUP (Density, CHOOSE (FiberDepthCondition, HardRockStructure, SoftRockStructure, NormalStructure) , 4)))	
AP	MainFeederAerialCopperStructure	Main Feeder Aerial Copper Structure $		= IF (NumberMainFdrPoles = 0, 0, CopperMainFdrStructureRatio * WaterCostMultiplier * NumberMainFdrPoles * VLOOKUP (Density, CHOOSE (FiberDepthCondition, HardRockStructure, SoftRockStructure, NormalStructure) , 6))	
AQ	TotalMainFdrCopperStructure	Total Main Feeder Copper Structure $		= MainFeederUGCopperStructure + MainFeederBuriedCopperStructure + MainFeederAerialCopperStructure	
AR	TotalMainFdrCopperCost	Total Main Feeder Copper Cost		= TotalMainFdrCopperMaterial + TotalMainFdrCopperStructure	
AS	CopperMainFeederCostPerLine	Copper Main Feeder Cost Per Line		= IF (MainFdrCopperWrkgLines = 0, 0, TotalMainFdrCopperCost / CopperMainFdrPrsNeeded)	
AT	CumulativeMainFdrCopperCostperPair	Cumulative Main Feeder Copper Cost per Line		= IF (CableTestColumn = "None", 0, SUMIF (OFFSET (CableTestColumn, 0, 0, - (FeederSegmentSequenceNumber)) , CLLIQuad, OFFSET (CopperMainFeederCostPerLine, 0, 0, - (FeederSegmentSequenceNumber) ,)) - SUMIF (OFFSET (CableBranchTestColumn, 0, 0, - FeederSegmentSequenceNumber) , CLLISegment, OFFSET (CopperMainFeederCostPerLine, 0, 0, - FeederSegmentSequenceNumber)))	
AU	MainFeederUGFiberStructure	Main Feeder Underground Fiber Structure $		= IF (MainFdrSegmentUGDistance = 0, 0, FiberMainFdrStructureRatio * WaterCostMultiplier * (IF (NumberMainFeederDucts < = 9, MainFeederStandardManholeCost, MainFeederExpandedManholeCost) * NumberMainFdrManholes + MainFdrSegmentUGDistance * (NumberMainFeederDucts * VLOOKUP (0, ConduitDensityTable, DensityColumnPointer) + VLOOKUP (Density, CHOOSE (FiberDepthCondition, HardRockStructure, SoftRockStructure, NormalStructure) , 2))))	
AV	MainFeederBuriedFiberStructure	Main Feeder Buried Fiber Structure $		= IF ('MainFdrSegmentBurDistance' = 0, 0, FiberMainFdrStructureRatio * WaterCostMultiplier * 'MainFdrSegmentBurDistance' * VLOOKUP (Density, CHOOSE (FiberDepthCondition, HardRockStructure, SoftRockStructure, NormalStructure) , 4))	
AW	MainFeederAerialFiberStructure	Main Feeder Aerial Fiber Structure $		= IF (NumberMainFdrPoles = 0, 0, FiberMainFdrStructureRatio * WaterCostMultiplier * NumberMainFdrPoles * VLOOKUP (Density, CHOOSE (FiberDepthCondition, HardRockStructure, SoftRockStructure, NormalStructure) , 6))	
AX	TotalMainFdrFiberStructure	Total Main Feeder Fiber Structure $		= MainFeederUGFiberStructure + MainFeederBuriedFiberStructure + MainFeederAerialFiberStructure	
AY	TotalMainFdrFiberCost	Total Main Feeder Fiber Cost		= TotalMainFdrFiberMaterial + TotalMainFdrFiberStructure	
AZ	MainFeederFiberCostPerFiber	Main Feeder Fiber Cost per Fiber		= IF (MainFdrWrkgFibers = 0, 0, TotalMainFdrFiberCost / MainFdrWrkgFibers)	The name associated with this cell is an absolute reference, so that the cumulation formulas wiill work
BA	CumulativeMainFdrFiberCostperFiber	Cumulative Main Feeder Fiber Cost per Fiber		= IF (FiberTestColumn = "None", 0, SUMIF (OFFSET (FiberTestColumn, 0, 0, - (FeederSegmentSequenceNumber)) , CLLIQuad, OFFSET (MainFeederFiberCostPerFiber, 0, 0, - (FeederSegmentSequenceNumber) ,)) - SUMIF (OFFSET (FiberBranchTestColumn, 0, 0, - FeederSegmentSequenceNumber) , CLLISegment, OFFSET (MainFeederFiberCostPerFiber, 0, 0, - FeederSegmentSequenceNumber)))	
BB	MainFeederUGCopperCostPerLine	Main Feeder Underground Copper Cost Per Line		= IF (MainFdrCopperWrkgLines = 0, 0, MainFdrSegmentUGCopperMaterial / CopperMainFdrPrsNeeded)	

BC	MainFeederBURCopperCostPerLine	Main Feeder Buried Copper Cost Per Line		= IF (MainFdrCopperWrkgLines = 0, 0, MainFdrSegmentBurCopperMaterial / CopperMainFdrPrsNeeded)	
BD	MainFeederAERCopperCostPerLine	Main Feeder Aerial Copper Cost Per Line		= IF (MainFdrCopperWrkgLines = 0, 0, MainFdrSegmentAerCopperMaterial / CopperMainFdrPrsNeeded)	
BE	MainFeederUGCopperStructureCostPerLine	Main Feeder Underground Copper Structure Cost Per Line		= IF (MainFdrCopperWrkgLines = 0, 0, MainFeederUGCopperStructure / CopperMainFdrPrsNeeded)	
BF	MainFeederBurCopperStructureCostPerLine	Main Feeder Buried Copper Structure Cost Per Line		= IF (MainFdrCopperWrkgLines = 0, 0, MainFeederBuriedCopperStructure / CopperMainFdrPrsNeeded)	
BG	MainFeederAerCopperStructureCostPerLine	Main Feeder Aerial Copper Structure Cost Per Line		= IF (MainFdrCopperWrkgLines = 0, 0, MainFeederAerialCopperStructure / CopperMainFdrPrsNeeded)	
BH	CumMainFdrUGCopperCostperLine	Cumulative Main Feeder Underground Copper Cost per Line		= IF (CableTestColumn = "None", 0, SUMIF (OFFSET (CableTestColumn, 0, 0, - (FeederSegmentSequenceNumber)), CLLIQuad, OFFSET (MainFeederUGCopperCostPerLine, 0, 0, - (FeederSegmentSequenceNumber) ,)) - SUMIF (OFFSET (CableBranchTestColumn, 0, 0, - FeederSegmentSequenceNumber) , CLLISegment, OFFSET (MainFeederUGCopperCostPerLine, 0, 0, - FeederSegmentSequenceNumber)))	
BI	CumMainFdrBurCopperCostperLine	Cumulative Main Feeder Buried Copper Cost per Line		= IF (CableTestColumn = "None", 0, SUMIF (OFFSET (CableTestColumn, 0, 0, - (FeederSegmentSequenceNumber)), CLLIQuad, OFFSET (MainFeederBurCopperCostPerLine, 0, 0, - (FeederSegmentSequenceNumber) ,)) - SUMIF (OFFSET (CableBranchTestColumn, 0, 0, - FeederSegmentSequenceNumber) , CLLISegment, OFFSET (MainFeederBurCopperCostPerLine, 0, 0, - FeederSegmentSequenceNumber)))	
BJ	CumMainFdrAERCopperCostperLine	Cumulative Main Feeder Aerial Copper Cost per Line		= IF (CableTestColumn = "None", 0, SUMIF (OFFSET (CableTestColumn, 0, 0, - (FeederSegmentSequenceNumber)), CLLIQuad, OFFSET (MainFeederAerCopperCostPerLine, 0, 0, - (FeederSegmentSequenceNumber) ,)) - SUMIF (OFFSET (CableBranchTestColumn, 0, 0, - FeederSegmentSequenceNumber) , CLLISegment, OFFSET (MainFeederAerCopperCostPerLine, 0, 0, - FeederSegmentSequenceNumber)))	
BK	CumMainFdrUGCopperStructureCostperLine	Cumulative Main Feeder Underground Copper Strucutre Cost per Line		= IF (CableTestColumn = "None", 0, SUMIF (OFFSET (CableTestColumn, 0, 0, - (FeederSegmentSequenceNumber)), CLLIQuad, OFFSET (MainFeederUGCopperStructureCostPerLine, 0, 0, - (FeederSegmentSequenceNumber) ,)) - SUMIF (OFFSET (CableBranchTestColumn, 0, 0, - FeederSegmentSequenceNumber) , CLLISegment, OFFSET (MainFeederUGCopperStructureCostPerLine, 0, 0, - FeederSegmentSequenceNumber)))	
BL	CumMainFdrBurCopperStructureCostperLine	Cumulative Main Feeder Buried Copper Structure Cost per Line		= IF (CableTestColumn = "None", 0, SUMIF (OFFSET (CableTestColumn, 0, 0, - (FeederSegmentSequenceNumber)), CLLIQuad, OFFSET (MainFeederBURCopperStructureCostPerLine, 0, 0, - (FeederSegmentSequenceNumber) ,)) - SUMIF (OFFSET (CableBranchTestColumn, 0, 0, - FeederSegmentSequenceNumber) , CLLISegment, OFFSET (MainFeederBURCopperStructureCostPerLine, 0, 0, - FeederSegmentSequenceNumber)))	
BM	CumMainFdrAerCopperStructureCostperLine	Cumulative Main Feeder Aerial Copper Structure Cost per Line		= IF (CableTestColumn = "None", 0, SUMIF (OFFSET (CableTestColumn, 0, 0, - (FeederSegmentSequenceNumber)), CLLIQuad, OFFSET (MainFeederAERCopperStructureCostPerLine, 0, 0, - (FeederSegmentSequenceNumber) ,)) - SUMIF (OFFSET (CableBranchTestColumn, 0, 0, - FeederSegmentSequenceNumber) , CLLISegment, OFFSET (MainFeederAERCopperStructureCostPerLine, 0, 0, - FeederSegmentSequenceNumber)))	
BN	MainFeederUGFiberCostPerFiber	Main Feeder Underground Fiber Cost Per Fiber		= IF (MainFdrWrkgFibers = 0, 0, MainFdrSegmentUGFiberMaterial / MainFdrWrkgFibers)	
BO	MainFeederBURFiberCostPerFiber	Main Feeder Buried Fiber Cost Per Fiber		= IF (MainFdrWrkgFibers = 0, 0, MainFdrSegmentBurFiberMaterial / MainFdrWrkgFibers)	
BP	MainFeederAERFiberCostPerFiber	Main Feeder Aerial Fiber Cost Per Fiber		= IF (MainFdrWrkgFibers = 0, 0, MainFdrSegmentAerFiberMaterial / MainFdrWrkgFibers)	

BQ	MainFeederUGFiberStructureCostPerFiber	Main Feeder Underground Fiber Structure Cost Per Fiber		= IF (MainFdrWrkgFibers = 0, 0, MainFeederUGFiberStructure / MainFdrWrkgFibers)	
BR	MainFeederBurFiberStructureCostPerFiber	Main Feeder Buried Fiber Structure Cost Per Fiber		= IF (MainFdrWrkgFibers = 0, 0, MainFeederBuriedFiberStructure / MainFdrWrkgFibers)	
BS	MainFeederAerFiberStructureCostPerFiber	Main Feeder Aerial Fiber Structure Cost Per Fiber		= IF (MainFdrWrkgFibers = 0, 0, MainFeederAerialFiberStructure / MainFdrWrkgFibers)	
BT	CumMainFdrUGFiberCostperFiber	Cumulative Main Feeder Underground Fiber Cost per Fiber		= IF (FiberTestColumn = "None", 0, SUMIF (OFFSET (FiberTestColumn, 0, 0, - (FeederSegmentSequenceNumber)) , CLLIQuad, OFFSET (MainFeederUGFiberCostPerfiber, 0, 0, - (FeederSegmentSequenceNumber) ,)) - SUMIF (OFFSET (FiberBranchTestColumn, 0, 0, - FeederSegmentSequenceNumber) , CLLISegment, OFFSET (MainFeederUGFiberCostPerfiber, 0, 0, - FeederSegmentSequenceNumber)))	
BU	CumMainFdrBurFiberCostperFiber	Cumulative Main Feeder Buried Fiber Cost per Fiber		= IF (FiberTestColumn = "None", 0, SUMIF (OFFSET (FiberTestColumn, 0, 0, - (FeederSegmentSequenceNumber)) , CLLIQuad, OFFSET (MainFeederBURFiberCostPerfiber, 0, 0, - (FeederSegmentSequenceNumber) ,)) - SUMIF (OFFSET (FiberBranchTestColumn, 0, 0, - FeederSegmentSequenceNumber) , CLLISegment, OFFSET (MainFeederBURFiberCostPerfiber, 0, 0, - FeederSegmentSequenceNumber)))	
BV	CumMainFdrAERFiberCostperFiber	Cumulative Main Feeder Aerial Fiber Cost per Fiber		= IF (FiberTestColumn = "None", 0, SUMIF (OFFSET (FiberTestColumn, 0, 0, - (FeederSegmentSequenceNumber)) , CLLIQuad, OFFSET (MainFeederAERFiberCostPerfiber, 0, 0, - (FeederSegmentSequenceNumber) ,)) - SUMIF (OFFSET (FiberBranchTestColumn, 0, 0, - FeederSegmentSequenceNumber) , CLLISegment, OFFSET (MainFeederAERFiberCostPerfiber, 0, 0, - FeederSegmentSequenceNumber)))	
BW	CumMainFdrUGFiberStructureCostperFiber	Cumulative Main Feeder Underground Fiber Strucutre Cost per Fiber		= IF (FiberTestColumn = "None", 0, SUMIF (OFFSET (FiberTestColumn, 0, 0, - (FeederSegmentSequenceNumber)) , CLLIQuad, OFFSET (MainFeederUGFiberStructureCostPerfiber, 0, 0, - (FeederSegmentSequenceNumber) ,)) - SUMIF (OFFSET (FiberBranchTestColumn, 0, 0, - FeederSegmentSequenceNumber) , CLLISegment, OFFSET (MainFeederUGFiberStructureCostPerfiber, 0, 0, - FeederSegmentSequenceNumber)))	
BX	CumMainFdrBurFiberStructureCostperFiber	Cumulative Main Feeder Buried Fiber Structure Cost per Fiber		= IF (FiberTestColumn = "None", 0, SUMIF (OFFSET (FiberTestColumn, 0, 0, - (FeederSegmentSequenceNumber)) , CLLIQuad, OFFSET (MainFeederBURFiberStructureCostPerfiber, 0, 0, - (FeederSegmentSequenceNumber) ,)) - SUMIF (OFFSET (FiberBranchTestColumn, 0, 0, - FeederSegmentSequenceNumber) , CLLISegment, OFFSET (MainFeederBURFiberStructureCostPerfiber, 0, 0, - FeederSegmentSequenceNumber)))	
BY	CumMainFdrAerFiberStructureCostperFiber	Cumulative Main Feeder Aerial Fiber Structure Cost per Fiber		= IF (FiberTestColumn = "None", 0, SUMIF (OFFSET (FiberTestColumn, 0, 0, - (FeederSegmentSequenceNumber)) , CLLIQuad, OFFSET (MainFeederAERFiberStructureCostPerfiber, 0, 0, - (FeederSegmentSequenceNumber) ,)) - SUMIF (OFFSET (FiberBranchTestColumn, 0, 0, - FeederSegmentSequenceNumber) , CLLISegment, OFFSET (MainFeederAERFiberStructureCostPerfiber, 0, 0, - FeederSegmentSequenceNumber)))	
BZ	IsSameAsPrevFeederBranch	Same As Previous Feeder Branch		= AND (IsSameAsPrevCLLIQuad, MainFeederBranch = prevMainFeederBranch)	
CA	IsSameAsPrevCLLIQuad	Same As Previous CLLI Quad		= (CLLIQuad = PrevCLLIQuad)	
CB	IsSameAsNextCLLIQuad	Same As Next CLLI Quad		= (CLLIQuad = NextCLLIQuad)	
CC	IsLastCLLIRecord	Last Record In CLLI		= OR (CLLI < > NextCLLI, NextCLLIQuad = "", ISBLANK (NextCLLIQuad))	
CD	IsFirstSegmentInBranch	First Segment In Branch		= NOT (IsSameAsPrevCLLIQuad)	
CE	QuadrantContainFiber*	Quadrant Contain Fiber		= IF (SegmentType1 = "Fiber", TRUE, (AND (IsSameAsNextCLLIQuad, NextQuadrantContainFiber = TRUE)))	

Sheet: Electronics&FDI

Col	Range Name	Column Name	Column Comment	Formula	Formula Comment
A	DLCLFiberElectronics	DLC-L On Fiber Electronics $		= IF (TotalLinesServedInGrid = 0, 0, CHOOSE (GridPlantIndex, 0, 0, DLC_LDiscount * (NumberDLCLsatFullCapacity * VLOOKUP (673, DigitalCarrierCost, 2) + NumberDLCLsatFullCapacity * 1344 * VLOOKUP (673, DigitalCarrierCost, 3) + VLOOKUP (SwitchedLinesinOverflowDLCLTerminal, DigitalCarrierCost, 2) + SwitchedLinesinOverflowDLCLTerminal * VLOOKUP (SwitchedLinesinOverflowDLCLTerminal, DigitalCarrierCost, 3))))	
B	GridLinesonDLCL	Grid Lines Served by DLC-L	Grid Lines served by DLC-L	= IF (GridPlantIndex = 3, VoiceGradeLinesEquippedatTerminal, 0)	
C	TotalCLLILineson DLCL	Total CLLI Lines Served by DLC-L	Total CLLI Lines served by DLC-L	= SUMIF (OFFSET (GridDataFirstRow, 0, 0, LastDataRow, 1) , CLLI, OFFSET (FirstGridLinesOnDLCL, 0, 0, LastDataRow, 1))	
D	COTLElectronics	COT-L Electronics $	Central Office DLC-L Terminal Investment	= IF (TotalCLLILinesOnDLCL = 0, 0, CHOOSE (GridPlantIndex, 0, 0, DLC_LDiscount * ((INT (TotalCLLILinesOnDLCL / MaxCOTDLCL) * VLOOKUP (MaxCOTDLCL, DLCCOTInvestment, 2) + VLOOKUP (MAX (MOD (TotalCLLILinesOnDLCL, MaxCOTDLCL) , 673) , DLCCOTInvestment, 2))) / TotalCLLILinesOnDLCL + COTDLCLPerLine) * GridLinesOnDLCL)	
E	DLCSFiberElectronics	DLC-S On Fiber Electronics $		= IF (TotalLinesServedInGrid = 0, 0, CHOOSE (GridPlantIndex, 0, DLC_SDiscount * (VLOOKUP (VoiceGradeLinesEquippedatTerminal, DigitalCarrierCost, 2) + VoiceGradeLinesEquippedatTerminal * VLOOKUP (VoiceGradeLinesEquippedatTerminal, DigitalCarrierCost, 3)) , 0))	
F	GridLinesonDLCS	Grid Lines Served by DLC-S	Grid Lines served by DLC-L	= IF (GridPlantIndex = 2, VoiceGradeLinesEquippedatTerminal, 0)	
G	TotalCLLILineson DLCS	Total CLLI Lines Served by DLC-S	Total CLLI Lines served by DLC-L	= SUMIF (OFFSET (GridDataFirstRow, 0, 0, LastDataRow, 1) , CLLI, OFFSET (FirstGridLinesOnDLCS, 0, 0, LastDataRow, 1))	
H	COTSElectronics	COT-S Electronics $	Central Office DLC-L Terminal Investment	= IF (TotalCLLILinesOnDLCS = 0, 0, CHOOSE (GridPlantIndex, 0, DLC_SDiscount * (CEILING ((TotalCLLILinesOnDLCS / MaxCOTDLCS) , 1) * VLOOKUP (MaxCOTDLCS, DLCCOTInvestment, 2) / TotalCLLILinesOnDLCS + COTDLCSPerLine) , 0) * GridLinesOnDLCS)	
I	CopperDS1Electronics	Copper DS1 Electronics		= CHOOSE (GridPlantIndex, DLC_SDiscount * CEILING (GridLinesProvisionedAsDS1s / 24, 1) * CopperT1, 0, 0)	
J	FiberDS1Electronics	Fiber DS1 Electronics		= DLC_LDiscount * DS3DS1UnitsperTerminalLocation * OpticsCost	
K	TotalElectronics	Total Electronics		= DLCLFiberElectronics + DLCSFiberElectronics + CopperDS1Electronics + FiberDS1Electronics + COTSElectronics + COTLElectronics	
L	MaxSizeFDICost	Max Size FDI Cost		= VLOOKUP (MaxSizeFDI, OutdoorSAICostTable, DensityColumnPointer)	
M	FDISizingFix	Temporary FDI Sizing Fix		= IF (ULQFDITotalPairs + LLQFDITotalPairs + URQFDITotalPairs + LRQFDITotalPairs = 0, 0, IF (ULQFDITotalPairs + LLQFDITotalPairs + URQFDITotalPairs + LRQFDITotalPairs < 600, 1 / (IF (ULQFDITotalPairs > 0, 1, 0) + IF (LLQFDITotalPairs > 0, 1, 0) + IF (URQFDITotalPairs > 0, 1, 0) + IF (LRQFDITotalPairs > 0, 1, 0)) , IF (ULQFDITotalPairs + LLQFDITotalPairs + URQFDITotalPairs + LRQFDITotalPairs < 1200, 2 / MAX (2, (IF (ULQFDITotalPairs > 0, 1, 0) + IF (LLQFDITotalPairs > 0, 1, 0) + IF (URQFDITotalPairs > 0, 1, 0) + IF (LRQFDITotalPairs > 0, 1, 0))) , 1)))	
N	ULQFDITotalPairs	ULQ FDI Total Pairs		= ULQResidualCableSize + ULQMaxSizeConnectingCables * MaxFeederSize + 2 * (ULQMaxSizeBackboneCables * MaxDistrSize + ULQResidualBackboneCableSize)	
O	ULQFDIsatMaxCapacity	ULQ FDIs At Max Capacity		= IF (ULQFDITotalPairs > MaxSizeFDI, TRUNC (ULQFDITotalPairs / MaxSizeFDI, 1) , 0)	
P	ResidualULQFDIPairs	Residual ULQ FDI Pairs		= ULQFDITotalPairs - (MaxSizeFDI * ULQFDIsatMaxCapacity)	
Q	ULQFDICost	ULQ FDI Cost		= FDISizingFix * IF (ULQFDITotalPairs = 0, 0, VLOOKUP (ResidualULQFDIPairs, OutdoorSAICostTable, DensityColumnPointer) + ULQFDIsatMaxCapacity * MaxSizeFDICost)	
R	LLQFDITotalPairs	LLQ FDI Total Pairs		= LLQResidualCableSize + LLQMaxSizeConnectingCables * MaxFeederSize + 2 * (LLQMaxSizeBackboneCables * MaxDistrSize + LLQResidualBackboneCableSize)	
S	LLQFDIsatMaxCapacity	LLQ FDIs At Max Capacity		= IF (LLQFDITotalPairs > MaxSizeFDI, TRUNC (LLQFDITotalPairs / MaxSizeFDI, 1) , 0)	

T	ResidualLLQFDIPairs	Residual LLQ FDI Pairs		= LLQFDITotalPairs - (MaxSizeFDI * LLQFDIsatMaxCapacity)	
U	LLQFDICost	LLQ FDI Cost		= FDISizingFix * IF (LLQFDITotalPairs = 0, 0, VLOOKUP (ResidualLLQFDIPairs, OutdoorSAICostTable, DensityColumnPointer) + LLQFDIsatMaxCapacity * MaxSizeFDICost)	
V	URQFDITotalPairs	URQ FDI Total Pairs		= URQResidualCableSize + URQMaxSizeConnectingCables * MaxFeederSize + 2 * (URQMaxSizeBackboneCables * MaxDistrSize + URQResidualBackboneCableSize)	
W	URQFDIsatMaxCapacity	URQ FDIs At Max Capacity		= IF (URQFDITotalPairs > MaxSizeFDI, TRUNC (URQFDITotalPairs / MaxSizeFDI, 1) , 0)	
X	ResidualURQFDIPairs	Residual URQ FDI Pairs		= URQFDITotalPairs - (MaxSizeFDI * URQFDIsatMaxCapacity)	
Y	URQFDICost	URQ FDI Cost		= FDISizingFix * IF (URQFDITotalPairs = 0, 0, VLOOKUP (ResidualURQFDIPairs, OutdoorSAICostTable, DensityColumnPointer) + URQFDIsatMaxCapacity * MaxSizeFDICost)	
Z	LRQFDITotalPairs	LRQ FDI Total Pairs		= LRQResidualCableSize + LRQMaxSizeConnectingCables * MaxFeederSize + 2 * (LRQMaxSizeBackboneCables * MaxDistrSize + LRQResidualBackboneCableSize)	
AA	LRQFDIsatMaxCapacity	LRQ FDIs At Max Capacity		= IF (LRQFDITotalPairs > MaxSizeFDI, TRUNC (LRQFDITotalPairs / MaxSizeFDI, 1) , 0)	
AB	ResidualLRQFDIPairs	Residual LRQ FDI Pairs		= LRQFDITotalPairs - (MaxSizeFDI * LRQFDIsatMaxCapacity)	
AC	LRQFDICost	LRQ FDI Cost		= FDISizingFix * IF (LRQFDITotalPairs = 0, 0, VLOOKUP (ResidualLRQFDIPairs, OutdoorSAICostTable, DensityColumnPointer) + LRQFDIsatMaxCapacity * MaxSizeFDICost)	
AD	FeederDistributionInterfaceCost	Feeder Distribution Interface Cost		= ULQFDICost + LLQFDICost + URQFDICost + LRQFDICost	
AE	AllocationFDItoAerialCopper	Allocation Of FDI To Aerial Copper		= FeederDistributionInterfaceCost * VLOOKUP (Density, CHOOSE (CopperDepthCondition, DistrHardMixTable, DistrSoftMixTable, DistrNormMixTable) , 4)	
AF	AllocationFDItoBuriedCopper	Allocation Of FDI To Buried Copper		= FeederDistributionInterfaceCost * VLOOKUP (Density, CHOOSE (CopperDepthCondition, DistrHardMixTable, DistrSoftMixTable, DistrNormMixTable) , 3)	
AG	AllocationFDItoUGCopper	Allocation Of FDI To UG Copper		= FeederDistributionInterfaceCost * VLOOKUP (Density, CHOOSE (CopperDepthCondition, DistrHardMixTable, DistrSoftMixTable, DistrNormMixTable) , 2)	

Sheet: Main&SubfeederAllocation

Col	Range Name	Column Name	Column Comment	Formula	Formula Comment
A	AllocationAerialCopperCable	Allocation Of Aerial Copper Cable		= (CumMainFdrAerCopperCostperLine + CumSubFdrAERCopperCostperLine) * SubfeederCopperPairsthisGrid	
B	AllocationBuriedCopperCableinclStructure	Allocation Of Buried Copper Cable Incl Structure		= (CumMainFdrBurCopperCostperLine + CumSubFdrBURCopperCostperLine + CumMainFdrBurCopperStructureCostperLine + CumSubFdrBURCopperStrucutreCostperLine) * SubfeederCopperPairsthisGrid	
C	AllocationUGCopperCable	Allocation Of UG Copper Cable		= (CumMainFdrUGCopperCostperLine + CumSubFdrUgCopperCostperLine) * SubfeederCopperPairsthisGrid	
D	AllocationCopperrelatedpoles	Allocation Of Copper Related Poles		= (CumMainFdrAerCopperStructureCostperLine + CumSubFdrAERCopperStructureCostperLine) * SubfeederCopperPairsthisGrid	
E	AllocationCopperRelatedConduit	Allocation Of Copper Related Conduit		= (CumMainFdrUGCopperStructureCostperLine + CumSubFdrUGCopperStrucutreCostperLine) * SubfeederCopperPairsthisGrid	
F	TotCopperFdrAttributoGrid	Total Copper Feeder Plant Attributed to This Grid		= AllocationAerialCopperCable + AllocationBuriedCopperCableinclStructure + AllocationUGCopperCable + AllocationCopperRelatedPoles + AllocationCopperRelatedConduit	
G	AllocationAerialFiberCable	Allocation Of Aerial Fiber Cable		= IF (TotalDLCSLinesinSubFeeder = 0, 0, IF (GridPlantType = "DLC - S", (NumberDLCSFibersinSubfeeder * (TotalLinesServedInGrid / ElectronicFill) / TotalDLCSLinesinSubFeeder) , NumberFibersPerGrid) * (CumMainFdrAERFiberCostperFiber)) + IF (GridPlantType = "DLC - S", (TotalLinesServedInGrid / ElectronicFill) * (CumDLCSSubFdrAERFiberCostperLine) , CumSubFdrAERFiberCostperFiber * NumberFibersPerGrid)	
H	AllocationBuriedFiberCableinclStructure	Allocation Of Buried Fiber Cable Incl Structure		= IF (TotalDLCSLinesinSubFeeder = 0, 0, IF (GridPlantType = "DLC - S", (NumberDLCSFibersinSubfeeder * (TotalLinesServedInGrid / ElectronicFill) / TotalDLCSLinesinSubFeeder) , NumberFibersPerGrid) * (CumMainFdrBURFiberCostperFiber + CumMainFdrBURFiberStructureCostperFiber)) + IF (GridPlantType = "DLC - S", (TotalLinesServedInGrid / ElectronicFill) * (CumDLCSSubFdrBURFiberCostperLine + CumDLCSSubFdrBURFiberStrucutreCostperLine) , (CumSubFdrBURFiberCostperFiber + CumSubFdrBURFiberStrucutreCostperFiber) * NumberFibersPerGrid)	
I	AllocationUGFiberCable	Allocation Of UG Fiber Cable		= IF (TotalDLCSLinesinSubFeeder = 0, 0, IF (GridPlantType = "DLC - S", (NumberDLCSFibersinSubfeeder * (TotalLinesServedInGrid / ElectronicFill) / TotalDLCSLinesinSubFeeder) , NumberFibersPerGrid) * (CumMainFdrUGFiberCostperFiber)) + IF (GridPlantType = "DLC - S", (TotalLinesServedInGrid / ElectronicFill) * (CumDLCSSubFdrUGFiberCostperLine) , CumSubFdrUGFiberCostperFiber * NumberFibersPerGrid)	
J	AllocationFiberrelatedpoles	Allocation Of Fiber Related Poles		= IF (TotalDLCSLinesinSubFeeder = 0, 0, IF (GridPlantType = "DLC - S", (NumberDLCSFibersinSubfeeder * (TotalLinesServedInGrid / ElectronicFill) / TotalDLCSLinesinSubFeeder) , NumberFibersPerGrid) * (CumMainFdrAERFiberStructureCostperFiber)) + IF (GridPlantType = "DLC - S", (TotalLinesServedInGrid / ElectronicFill) * (CumDLCSSubFdrAERFiberStrucutreCostperLine) , CumSubFdrAERFiberStrucutreCostperFiber * NumberFibersPerGrid)	
K	AllocationFiberRelatedConduit	Allocation Of Fiber Related Conduit		= IF (TotalDLCSLinesinSubFeeder = 0, 0, IF (GridPlantType = "DLC - S", (NumberDLCSFibersinSubfeeder * (TotalLinesServedInGrid / ElectronicFill) / TotalDLCSLinesinSubFeeder) , NumberFibersPerGrid) * (CumMainFdrUGFiberStructureCostperFiber)) + IF (GridPlantType = "DLC - S", (TotalLinesServedInGrid / ElectronicFill) * (CumDLCSSubFdrUGFiberStrucutreCostperLine) , CumSubFdrUGFiberStrucutreCostperFiber * NumberFibersPerGrid)	
L	TotFiberFdrAttributoGrid	Total Fiber Feeder Plant Attributable to This Grid		= AllocationAerialFiberCable + AllocationBuriedFiberCableinclStructure + AllocationUGFiberCable + AllocationFiberRelatedPoles + AllocationFiberRelatedConduit	
M	TotalDLCLonFiber	Total DLC-L Electronics $		= DLCLFiberElectronics + COTLElectronics	
N	TotalDLCSonFiber	Total DLC-S Electronics $		= DLCSFiberElectronics + COTSElectronics	

O	SubtotalFeederCost	Subtotal Feeder Cost		= IF (TotalLinesServedInGrid = 0, 0, TotCopperFdrAttribtoGrid + TotFiberFdrAttribtoGrid + TotalDLCLonFiber + TotalDLCSonFiber + CopperDS1Electronics + FiberDS1Electronics)	
P	GrandTotalFeeder Cost	Grand Total Feeder Cost		= SubtotalFeederCost + TotalSubfeederPart2 + TotalDLCtoFDICableCost + FeederDistributionInterfaceCost	
Q	TotalMainSubfeed erPoleCost	Total Main/Subfeeder Pole Cost		= AllocationCopperRelatedPoles + AllocationFiberRelatedPoles	
R	TotalMainSubfeed erConduitCost	Total Main/Subfeeder Conduit Cost		= AllocationCopperRelatedConduit + AllocationFiberRelatedConduit	

Sheet: Switching

Col	Range Name	Column Name	Column Comment	Formula	Formula Comment
A	SwitchingCLLI	CLLI		= CLLI	
B	SwitchingSwitched LinesEquipped	Switched Lines Equipped		= SwitchedLinesInGrid	
C	TotalLinesEquipped InSwitch	Total Lines Equipped In Switch		= SUMIF (OFFSET (SwitchingCLLI, 0, 0, LastDataRow, 1) , CLLI, OFFSET (SwitchingSwitchedLinesEquipped, 0, 0, LastDataRow, 1))	
D	SwitchingFixedCostPerLine	Fixed Cost/Ln		= IF (TotalLinesEquippedInSwitch = 0, 0, (VLOOKUP (CompanySize, COSwitchCostTable, 2) * TrfSen) / TotalLinesEquippedInSwitch * (1 + VLOOKUP (CompanySize, COSwitchCostTable, 6)))	
E	SwitchingVariable CostPerLine	Variable Cost Per Line		= VLOOKUP (CompanySize, COSwitchCostTable, 3) / SwitchFillFactor * TrfSen * (1 + VLOOKUP (CompanySize, COSwitchCostTable, 6))	
F	SwitchingTotalLineCost	Total Line Cost		= SwitchingFixedCostPerLine + SwitchingVariableCostPerLine	
G	SwitchingTotalInvestment	Total Switch Investment		= SwitchedLinesInGrid * SwitchingTotalLineCost	

Sheet: Investment

Col	Range Name	Column Name	Column Comment	Formula	Formula Comment
A	AerialCopperCable Investment	Aerial Copper Cable Investment $		= AllocationAerialCopperCable + SubfeederPt2AerialCopperMaterial + TotalDLCFDIAerialCableMaterial + AllocationFDIToAerialCopper + TotalAerialCableDistributionCost + TotalNIDCostAllocatedAerial	
B	BuriedCopperCable Investment	Buried Copper Cable Investment $		= AllocationBuriedCopperCableinclStructure + SubfeederPt2BuriedCopperMaterial + SubfeederPt2BuriedCopperStructure + TotalDLCFDIBuriedCableMaterial + TotalBuriedDLCFDICableStructure + AllocationFDIToBuriedCopper + TotalBuriedCableDistributionCost + TotalNIDCostAllocatedBuried	
C	UndergroundCopperCableInvestment	Underground Copper Cable Investment $		= AllocationUGCopperCable + SubfeederPt2UGCopperMaterial + TotalDLCFDIUGCableMaterial + AllocationFDIToUGCopper + TotalUGCableDistributionCost	
D	PoleInvestmentAssignedCopper	Pole Investment Assigned Copper		= AllocationCopperRelatedPoles + TotalAerialDLCFDICableStructure + IF (GridPlantType = "Cable", SubfeederPt2AerialStructure, 0) + TotalPoleDistributionCost	
E	ConduitInvestment AssignedCopper	Conduit Investment Assigned Copper		= AllocationCopperRelatedConduit + TotalUGDLCFDICableStructure + IF (GridPlantType = "Cable", SubfeederPt2UGStructure, 0) + TotalDistributionConduitCost	
F	CopperRelatedPlantTotal	Copper Related Plant Total		= AerialCopperCableInvestment + BuriedCopperCableInvestment + UndergroundCopperCableInvestment + PoleInvestmentAssignedCopper + Conduit Investment Assigned Copper	
G	AerialFiberCableInvestment	Aerial Fiber Cable Investment $		= AllocationAerialFiberCable + SubfeederPt2AerialFiberMaterial	
H	BuriedFiberCableInvestment	Buried Fiber Cable Investment $		= AllocationBuriedFiberCableinclStructure + SubfeederPt2BuriedFiberMaterial + SubfeederPt2BuriedFiberStructure	
I	UndergroundFiber CableInvestment	Underground Fiber Cable Investment $		= AllocationUGFiberCable + SubfeederPt2UGFiberMaterial	
J	PoleInvestmentAssignedFiber	Pole Investment Assigned Fiber		= AllocationFiberRelatedPoles + IF (GridPlantType = "Cable", 0, SubfeederPt2AerialStructure)	
K	ConduitInvestment AssignedFiber	Conduit Investment Assigned Fiber		= AllocationFiberRelatedConduit + IF (GridPlantType = "Cable", 0, SubfeederPt2UGStructure)	
L	TotalFiberRelatedPlant	Total Fiber Related Plant		= AerialFiberCableInvestment + BuriedFiberCableInvestment + UndergroundFiberCableInvestment + PoleInvestmentAssignedFiber + ConduitInvestmentAssignedFiber	

Sheet: Output

Col	Range Name	Column Name	Column Comment	Formula	Formula Comment
A		Clli		= CLLI	
B		State Id		= StateId	
C		Company Size		= CompanySize	
D		Density		= DensityColumnPointer	
E		FDI Code		= FDICode	
F		Lines Times Average Distribution Length		= AverageDistributionDistance * TotalWorkingLinesServedInGrid	
G		Lines Times Average Feeder Length		= AverageFeederDistance * TotalWorkingLinesServedInGrid	
H		Area-sq Miles		= Area_sq_Miles	
I		Capped Cost Factor		= IF (UncappedTotalLoopCost = 0, 1, CappedTotalLoopCost / UncappedTotalLoopCost)	
J		Residence Lines		= ResidenceLines	
K		Single Business Lines		= SingleBusinessLines	
L		Total House Holds		= TotalHouseholds	
M		Total Business Lines		= SingleBusinessLines + MultiBusinessLines	
N		Total Working Lines Served in Grid		= TotalWorkingLinesServedInGrid	
O	AverageLoopLength	Average Loop Length		= IF (AverageFeederDistance = 10000, 0, AverageFeederDistance) + AverageDistributionDistance	
P		Total Lines Loop Length		= TotalWorkingLinesServedInGrid * AverageLoopLength	
Q		Total Distribution Cost		= TotalDistributionCost	
R		Grand Total Feeder Cost		= GrandTotalFeederCost	
S	UncappedTotalLoopCost	Uncapped Total Loop Cost		= TotalDistributionCost + GrandTotalFeederCost	
T	ActualLoopCostPerLine	Actual Loop Cost Per Line		= IF (TotalWorkingLinesServedInGrid = 0, 0, (UncappedTotalLoopCost / TotalWorkingLinesServedInGrid))	
U	CappedLoopCostPerLine	Capped Loop Cost Per Line		= IF (ActualLoopCostPerLine > InvestmentLoopCap, InvestmentLoopCap, ActualLoopCostPerLine)	
V	CappedTotalLoop Cost	Capped Total Loop Cost		= CappedLoopCostPerLine * TotalWorkingLinesServedInGrid	
W		2111 Land Investment		= SwitchingTotalInvestment * SwLandFactor	
X		2121 Building Investment		= SwitchingTotalInvestment * SwBldgFactor	
Y		2210 Central Office Switching Investment		= SwitchingTotalInvestment	

Z	TotalDLCDS1Investment	Uncapped 2230 Total DLC, DS1 Electronic Investment		= TotalElectronics	
AA	TotalInterOfficeFacilities	2230 Uncapped Interoffice Facilities		= SwitchingTotalInvestment * InterofficeSwRatio	
AB		Uncapped Total Circuit		= TotalDLCDS1Investment + TotalInterOfficeFacilities	
AC		UnCapped NID Cost		= TotalGridNIDCost	
AD		UnCapped 2421 Aerial Copper Cable Investment		= AerialCopperCableInvestment	
AE		UnCapped 2422 Underground Copper Cable Investment		= UndergroundCopperCableInvestment	
AF		UnCapped 2423 Buried Copper Cable Investment		= BuriedCopperCableInvestment	
AG		UnCapped 2421 Aerial Fiber Cable Investment		= AerialFiberCableInvestment	
AH		UnCapped 2422 Underground Fiber Cable Investment		= UndergroundFiberCableInvestment	
AI		UnCapped 2423 Buried Fiber Cable Investment		= BuriedFiberCableInvestment	
AJ		UnCapped 2411 Pole Line Investment		= PoleInvestmentAssignedCopper + PoleInvestmentAssignedFiber	
AK		UnCapped 2441 Conduit Investment		= ConduitInvestmentAssignedCopper + ConduitInvestmentAssignedFiber	
AL		Aerial Route Length		= MainFdrSegmentAerDistance + SubfeederAERDistance + Pt2AerialCopperDistance + Pt2AerialFiberDistance + TotalAerialDLCFDICableDistance + TotalAerialCableDistributionDistance	
AM		Buried Route Length		= MainFdrSegmentBurDistance + SubfeederBURDistance + Pt2BuriedCopperDistance + Pt2BuriedFiberDistance + TotalburiedDLCFDICableDistance + TotalBuriedCableDistributionDistance	
AN		Underground Route Length		= MainFdrSegmentUGDistance + SubfeederUGDistance + Pt2UGCopperDistance + Pt2UGFiberDistance + ULQUGCableDistance + LLQUGCableDistance + LeftSideCommonUGCableDistance + URQUGCableDistance + LRQUGCableDistance + RightSideCommonUGCableDistance + ULQBranchBackboneUGDistance * (NumberULQbranchcables + 1 / 2) + LLQBranchBackboneUGDistance * (NumberLLQbranchcables + 1 / 2) + URQBranchBackboneUGDistance * (NumberURQbranchcables + 1 / 2) + LRQBranchBackboneUGDistance * (NumberLRQbranchcables + 1 / 2)	
AO		Number of Poles		= NumberMainFdrPoles + NumberSubfeederPoles + ((Pt2AerialCopperDistance + Pt2AerialFiberDistance) / VLOOKUP (Density, FeederSpacingTable, 3)) + NumberofDLCFDIPoles + NumberPolesForDistribution	

AP		Number of Manholes		= NumberMainFdrManholes + NumberSubfeederManholes + NumberSubFdrPt2Manholes + URQManholes + LRQManholes + RightSideCommonManholes + ULQManholes + LLQManholes + LeftSideCommonManholes + LRQBackboneManholes + LRQBranchManholes + URQBackboneManholes + URQBranchManholes + LLQBackboneManholes + LLQBranchManholes + ULQBackboneManholes + ULQBranchManholes	
AQ		Grid Lines Served on DLC-L		= IF (GridPlantType = "DLC - L", TotalWorkingLinesServedInGrid, 0)	
AR		Grid Lines Served on DLC-S		= IF (GridPlantType = "DLC - S", TotalWorkingLinesServedInGrid, 0)	
AS		Grid Lines Served on Copper		= IF (GridPlantType = "Cable", TotalWorkingLinesServedInGrid, 0)	
AT		Number of DLC-L Terminals		= IF (GridPlantType = "DLC - L", NumberDLCLsatFullCapacity + IF (SwitchedLinesinOverflowDLCLTerminal > 0, 1, 0) , 0)	
AU		Number of DLC-S Terminals		= IF (GridPlantType = "DLC - S", 1, 0)	

UNIVERSAL SERVICE, DOCKET NO. 97-00888
PHASE II - COST ISSUES
RESPONSE TO ISSUE NO. 16

Question 16 (e) What method should be used to determine structure sharing (e.g., poles, trenches, conduits)?

Response: These percentages should be calculated in the same manner as the plant mix percentages: by persons familiar with both standard engineering practices and the specific area in question. Again, most probably the engineers and or costing managers responsible for structure sharing (to the extent that it exists) within the LECs serving the area today. Also, as in the case of plant mix, the forward-looking nature of the model would need to be kept foremost in mind when calculating percentages. However, the actual experience of these individuals should serve as the foundation for the percentage estimates.

UNIVERSAL SERVICE, DOCKET NO. 97-00888
PHASE II - COST ISSUES
RESPONSE TO ISSUE NO. 16

Question 16 (f) What method should be used to determine the most economically efficient fiber-copper cross-over point?

Response: The method should NOT be based solely on which crossover point provides the least cost solution because other factors, such as quality and speed of transmission, must be considered when calculating this value. This point should be determined by persons with extensive outside plant engineering experience and personal knowledge of the area served as well as personal knowledge of accepted transmission standards and what is required to equate transmission speed and quality across all customer locations. This point should also be adjustable to take into consideration different gauges of cable which can affect the relative efficiency of various crossover points.

UNIVERSAL SERVICE, DOCKET NO. 97-00888
PHASE II - COST ISSUES
RESPONSE TO ISSUE NO. 16

Question 16 (g) What loop design standards, if any, should be adopted for the cost model?

Response: The loop design standards that should be adopted are those that are outlined on pages 32-43 of the document Model Methodology attached. Specifically found on page 34 are the requirements regarding maximum loop length for each carrier serving area, as well as standards dealing with the option of loaded loops (which should be avoided, per the FCC's mandates found in the May 7th Order on Universal Service). As stated in the Model Methodology, the constraints listed on page 34 ensure compliance with standard AT&T/Lucent and US practices covering loop resistance and dB loss.

UNIVERSAL SERVICE, DOCKET NO. 97-00888
PHASE II - COST ISSUES
RESPONSE TO ISSUE NO. 16

Question 16 (h) What size(s) of digital loop carriers should be the model incorporate?

Response: The model should incorporate the two sizes of DLC used in the BCPM3.0 which are described in detail in the Model Methodology document on page 38.

UNIVERSAL SERVICE, DOCKET NO. 97-00888
PHASE II - COST ISSUES
RESPONSE TO ISSUE NO. 16

Question 16 (i) What wireless threshold, if any, should the model use?

Response: The model should offer cost calculations two ways: Estimates produced using a wireless threshold and also estimates produced without one, so that comparisons can be made as to HOW efficient wireless might be as an alternative technology. The BCPM3.0 provides both types of costs, based on capped investment (using the wireless threshold) and uncapped investment. The wireless threshold in the model is set at investment of $10,000 per loop.

UNIVERSAL SERVICE, DOCKET NO. 97-00888
PHASE II - COST ISSUES
RESPONSE TO ISSUE NO. 16

Question 16 (j) What method should be used to determine the materials and installation costs of manholes, poles, anchors, guys, aerial cable, building attachments?

Response: These should be determined based on real world cost information obtained from LEC records of purchases and installations in the specific area being modeled. These costs should be allowed to (but not necessarily required to) vary by density zone, topography, and other dimensions as well as by geographic location. Obviously adjustments should be made (if necessary) to reflect the forward-looking nature of the model. Adjustments should also be made to reflect differences in the circumstances between the purchase/installation records and the scenarios used (and specific areas served) in the proxy model itself.

UNIVERSAL SERVICE, DOCKET NO. 97-00888
PHASE II - COST ISSUES
RESPONSE TO ISSUE NO. 16

Question 16 (k) What method should be used to determine NID costs?

Response: See response to j.

UNIVERSAL SERVICE, DOCKET NO. 97-00888
PHASE II - COST ISSUES
RESPONSE TO ISSUE NO. 16

Question 16 (l) What method should be used to determine the cost of investment and installation of service area interfaces?

Response: See response to j.

UNIVERSAL SERVICE, DOCKET NO. 97-00888
PHASE II - COST ISSUES
RESPONSE TO ISSUE NO. 16

Question 16(m) What method should be used to determine cable fill and utilization factors?

Response: The cable fill and utilization factors should be determined based on real world experience in the trigger points used for reinforcing the existing capacity with consideration of a reasonable allowance for defective cable pair and growth that would take place over the engineering and installation period.

UNIVERSAL SERVICE, DOCKET NO. 97-00888
PHASE II - COST ISSUES
RESPONSE TO ISSUE NO. 16

Question 16 (n) What method should be used to determine the mix of host, stand-alone, and remote switches?

Response: Please refer to the "Switching" chapter in the document Model Methodology pages 45-65.

UNIVERSAL SERVICE, DOCKET NO. 97-00888
PHASE II - COST ISSUES
RESPONSE TO ISSUE NO. 16

Question 16 (o) What switch capacity constraints, if any, should the model employ?

Response: See response to n.

UNIVERSAL SERVICE, DOCKET NO. 97-00888
PHASE II - COST ISSUES
RESPONSE TO ISSUE NO. 16

Question 16 (p) What method should be used to determine switching investment costs?

Response: See response to n.

UNIVERSAL SERVICE, DOCKET NO. 97-00888
PHASE II - COST ISSUES
RESPONSE TO ISSUE NO. 16

Question 16 (q) What method should be used to determine the portion of total interoffice trunking, signaling, and local tandem costs to be attributed to universal service?

Response: Please refer to the "Transport" chapter in the Model Methodology document pages 66-75.

UNIVERSAL SERVICE, DOCKET NO. 97-00888
PHASE II - COST ISSUES
RESPONSE TO ISSUE NO. 16

Question 16 (r) What method should be used to determine costs of general support facilities (e.g., vehicles, land, buildings)?

Response: The method that should be used is outlined in the document Model Methodology on page 90 and 91.

UNIVERSAL SERVICE, DOCKET NO. 97-00888
PHASE II - COST ISSUES
RESPONSE TO ISSUE NO. 16

Question 16 (s) What method should be used to determine the economic depreciation rate of assets?

Response: Depreciation rates should be based on economic lives, (as opposed to depreciation lives) and future net salvage percentages that fall within FCC authorized ranges, as prescribed in the FCC's May 7th Order on Universal Service (Order paragraph 250 (5).

UNIVERSAL SERVICE, DOCKET NO. 97-00888
PHASE II - COST ISSUES
RESPONSE TO ISSUE NO. 16

Question 16 (t) What method should be used to determine plant-specific (e.g., equipment and maintenance), non-plant-specific (e.g., engineering, network operations), customer service (e.g., marketing and billing), and corporate (e.g., legal and accounting) expense factors?

Response: The method that should be used to determine expense factors is outlined in the Model Methodology document on page 95. Specifically, operating expenses for investment accounts should be a function of that specific account. Actual expense data as a percentage of investment may be used as a basis for the expense factor. The percentage should of course be adjusted to reflect both the forward-looking nature of the model and the fact that expenses must reflect the provision of basic service only.

It is important to note that "forward-looking" adjustments may result in either increasing or decreasing the expense-as-percent-of-investment factor. While efficiency gains would tend to suggest lower expense factors, such possibilities as increased postage rates and increased labor rates may impact forward-looking expenses in a positive direction.

It is also important that the model allow the user to incorporate operating expenses on either a factor basis or a per-line basis. The BCPM3.0 allows the user the option of either or a combination of both of these methods.

UNIVERSAL SERVICE, DOCKET NO. 97-00888
PHASE II - COST ISSUES
RESPONSE TO ISSUE NO. 16

Question 16 (u) In which cases is it appropriate to allocate costs between the provision of universal service and all other services?

Response: Cost allocation, as a general concept, is inappropriate. Economists generally agree that the only situation in which "cost allocation" is appropriate is with regard to joint and common costs. Costs that are "caused" by the provision of basic service cannot be arbitrarily "allocated" to other services because they are not caused in the provision of those other services. The only costs that any model should "allocate" between basic service and other services are joint and common costs.

UNIVERSAL SERVICE, DOCKET NO. 97-00888
PHASE II - COST ISSUES
RESPONSE TO ISSUE NO. 16

Question 16 (v) In cases where it is appropriate, what method should be used to allocate costs between the provision of universal service and all other services?

Response: See response to u.

UNIVERSAL SERVICE, DOCKET NO. 97-00888
PHASE II - COST ISSUES
RESPONSE TO ISSUE NO. 16

Question 16 (w) What, if any, local usage component should be included in universal service support?

Response: Local usage (with regard to the switch and the percentage of switch activity that is modeled in the BCPM3.0 for basic service) should be unlimited. There is no limit on local usage that should be incorporated into the model because the FCC has not stated that a limit on local usage is appropriate for inclusion in universal service support.

UNIVERSAL SERVICE, DOCKET NO. 97-00888
PHASE II - COST ISSUES
RESPONSE TO ISSUE NO. 16

Question 16 (x) What is the proper cost and percentage of equity?

Response: The proper cost of equity is 12.91% and the proper percentage of equity is 67.2%, as used in the BCPM3.0.

UNIVERSAL SERVICE, DOCKET NO. 97-00888
PHASE II - COST ISSUES
RESPONSE TO ISSUE NO. 16

Question 16 (y) What is the proper cost of debt?

Response: The proper cost of debt is 7.8%, as used in the BCPM3.0.