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 (TotalHouseholdsColumn) + 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	ULQHousingUnits locations	ULQ Housing Units Locations		= IF (HousingUnits = 0, 0, TotalHousingUnitLocations * ULQHousingUnits / HousingUnits)	
D	ULQBusinesslocat ions	ULQ Business Locations		= IF (TotalBusinessLines = 0, 0, CEILING (BusinessLocations * ULQBusinessLines / TotalBusinessLines, 1))	
E	ULQLots	ULQ Lots		= ULQHousingUnitslocations + ULQBusinesslocations	
F	ULQLotSideLengt h	ULQ Lot Side Length		= IF (ULQLots = 0, 0, SQRT (ULQArea / ULQLots) * 5280)	
G	ULQQuadrantSide Length	ULQ Quadrant Side Length		= SQRT (ULQArea) * 5280	
H	PreULQBackbone andBranchCableL ength	Preliminary ULQ Backbone & Branch Cable Length		= IF (NumberpairsinULquadrant = 0, 0, IF (ULQQuadrantSideLength - (2 * ULQLotSideLength) < 0, 0, ULQQuadrantSideLength - (2 * ULQLotSideLength)))	
I	NumberULQbranc hcables	Number Of ULQ Branch Cables		= IF (OR (ULQLotSideLength = 0, PreULQBackboneandBranchCableLength = 0) , 0, CEILING (ULQQuadrantSideLength / (2 * ULQLotSideLength) , 1))	
J	ULQRoadLength Adj	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	ULQBackboneand BranchCableLengt h	ULQ Backbone &Branch Cable Length (Road Adjusted)		= PreULQBackboneandBranchCableLength * ULQRoadLengthAdj	
L	NumberpairsinUL quadrant	Number Of Pairs In UL Quadrant		= ULQHousingUnits * PairsPerHousingUnit + IF (ULQBusinesslocations = 0, 0, IF ((ULQBusinessLines / ULQBusinesslocations) > PairsPerBusinessLocation, ULQBusinessLines, PairsPerBusinessLocation * ULQBusinesslocations))	
M	LLQHousingUnitl ocations	LLQ Housing Unit Locations		= IF (HousingUnits = 0, 0, TotalHousingUnitLocations * LLQHousingUnits / HousingUnits)	
N	LLQBusinesslocat ions	LLQ Business Locations		= IF (TotalBusinessLines = 0, 0, CEILING (BusinessLocations * LLQBusinessLines / TotalBusinessLines, 1))	
O	LLQLots	LLQ Lots		= LLQHousingUnitlocations + LLQBusinesslocations	
P	LLQLotSideLengt h	LLQ Lot Side Length		= IF (LLQLots = 0, 0, SQRT (LLQArea / LLQLots) * 5280)	
Q	LLQQuadrantSide Length	LLQ Quadrant Side Length		= SQRT (LLQArea) * 5280	
R	PreLLQBackbone BranchCableLengt h	Preliminary LLQ Backbone & Branch Cable Length		= IF (NumberpairsinLLquadrant = 0, 0, IF (LLQQuadrantSideLength - (2 * LLQLotSideLength) < 0, 0, LLQQuadrantSideLength - (2 * LLQLotSideLength)))	
S	NumberLLQbranc hcables	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	URQRoadLength Adj	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	LRQQuadrantSide Length	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	TotalLoopPlanningLength	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 + AverageDLCtoFDIDistance	
CC	LargestQuadAreaNumber	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) > 11100, 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	LLQResidualCableSize	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) > 11100, 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	LLQBuriedCable Distance	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	LeftSideCommon CableResidualSize	Left Side Common Cable Residual Size		= INDEX (FeederCableSize, MATCH (LeftSideCommonCablePairsRequired - (MaxFeederSize * NumberLeftCommonMaxSizeCables) , FeederCableSize, - 1) , 1)	
AM	LeftSideCommon UGCableDistance	Left Side Common Underground Cable Distance		= IF (LeftSideCommonCablePairsRequired = 0, 0, VLOOKUP (Density, CHOOSE (CopperDepthCondition, DistrHardMixTable, DistrSoftMixTable, DistrNormMixTable) , 2) * LeftSideCommoncableDistance)	
AN	LeftSideCommon Ducts	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	LeftSideCommon StandardManhole Cost	Left Side Common Standard Manhole Cost		= IF (OR (LeftSideCommonManholes = 0, LeftSideCommonDucts > 9) , 0, VLOOKUP (LeftSideCommonDucts, CHOOSE (FiberDepthCondition, ManholeHardRockTable, ManholeSoftRockTable, ManholeNormalSurfaceTable) , DensityColumnPointer))	
AQ	LeftSideCommon ExpandedManhole Cost	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	LeftSideCommon ConduitCost	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	LeftSideCommon AerialcableDistan ce	Left Side Common Aerial Cable Distance		= IF (LeftSideCommonCablePairsRequired = 0, 0, VLOOKUP (Density, CHOOSE (CopperDepthCondition, DistrHardMixTable, DistrSoftMixTable, DistrNormMixTable) , 4) * LeftSideCommoncableDistance)	
AT	LeftSideCommon CableGauge	Left Side Common Cable Gauge		= MIN (LLQCableGauge, ULQCableGauge)	
AU	LeftSideCommon AerialcableMateri al	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	LeftSideCommon BuriedCableDista nce	Left Side Common Buried Cable Distance		= IF (LeftSideCommonCablePairsRequired = 0, 0, VLOOKUP (Density, CHOOSE (CopperDepthCondition, DistrHardMixTable, DistrSoftMixTable, DistrNormMixTable) , 3) * LeftSideCommoncableDistance)	
AW	LeftSideCommon BuriedCableMater ial	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	LeftSideCommon UGCableMaterial	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	RightSideCommo ncableDistance	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) > 11100, 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	URQBuriedCableDistance	URQ Buried Cable Distance		= VLOOKUP (Density, CHOOSE (CopperDepthCondition, DistrHardMixTable, DistrSoftMixTable, DistrNormMixTable) , 3) * URQConnectingCableDistance	
BO	URQBuriedCableMaterial	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, (LRQHorizontalfttoRoadCentroid + LRQverticalfttoRoadCentroid) * 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) > 11100, 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	LRQBuriedCable Distance	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	ULQBackboneUG CableMaterial	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	ULQBranchManh oles	Number ULQ Branch Manholes		= NumberULQbranchcables * IF (ULQBranchBackboneUGDistance < 1, 0, ULQBranchBackboneUGDistance / VLOOKUP (Density, DistrSpacingTable, 2))	
Q	ULQStandardBran chManholeCost	ULQ Standard Branch Manhole Cost		= IF (OR (ULQBranchManholes = 0, ULQBranchDucts > 9) , 0, VLOOKUP (ULQBranchDucts, CHOOSE (FiberDepthCondition, ManholeHardRockTable, ManholeSoftRockTable, ManholeNormalSurfaceTable) , DensityColumnPointer))	
R	ULQExpandedBra nchManholeCost	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	ULQBackboneDu cts	Number ULQ Backbone Ducts		= ULQMaxSizeBackboneCables + 2	
T	ULQBackboneMa nholes	Number ULQ Backbone Manholes		= IF (ULQBranchBackboneUGDistance < 1, 0, ULQBranchBackboneUGDistance / 2 / VLOOKUP (Density, DistrSpacingTable, 2))	
U	ULQStandardBac kboneManholeCos t	ULQ Standard Backbone Manhole Cost		= IF (OR (ULQBackboneManholes = 0, ULQBackboneDucts > 9) , 0, VLOOKUP (ULQBackboneDucts, CHOOSE (FiberDepthCondition, ManholeHardRockTable, ManholeSoftRockTable, ManholeNormalSurfaceTable) , DensityColumnPointer))	
V	ULQExpandedBac kboneManholeCos t	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	PairsperLLQbranc hcable	Pairs Per LLQ Branch Cable		= IF (NumberLLQbranchcables = 0, 0, CEILING (NumberpairsinLLquadrant / (2 * NumberLLQbranchcables) / DistributionFillFactor, 1))	
Y	LLQMaxSizeBran chCables	LLQ Max Size Branch Cables		= IF (PairsperLLQbranchcable > MaxDistrSize, TRUNC (PairsperLLQbranchcable / MaxDistrSize) , 0)	
Z	LLQResidualBran chCableSize	LLQ Residual Branch Cable Size		= IF (PairsperLLQbranchcable = 0, 0, INDEX (DistrCableSize, MATCH (PairsperLLQbranchcable - LLQMaxSizeBranchCables * MaxDistrSize, DistrCableSize, - 1) , 1))	
AA	LLQMaxSizeBack boneCables	LLQ Max Size Backbone Cables		= NumberLLQbranchcables * LLQMaxSizeBranchCables + IF (NumberLLQbranchcables * LLQResidualBranchCableSize > MaxDistrSize, TRUNC (NumberLLQbranchcables * LLQResidualBranchCableSize / MaxDistrSize) , 0)	
AB	LLQResidualBack boneCableSize	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	LLQBranchBackb oneAerialDistance	LLQ Branch & Backbone Aerial Distance		= VLOOKUP (Density, CHOOSE (CopperDepthCondition, DistrHardMixTable, DistrSoftMixTable, DistrNormMixTable) , 4) * LLQBackboneBranchCableLength	

AD	LLQBranchAerial CableMaterial	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	LLQBackboneBuriedCableMaterial	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	LLQBranchBackboneUGDistance	LLQ Branch & Backbone Underground Distance		= VLOOKUP (Density, CHOOSE (CopperDepthCondition, DistrHardMixTable, DistrSoftMixTable, DistrNormMixTable) , 2) * LLQBackboneBranchCableLength	
AJ	LLQBranchUGCableMaterial	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	LLQBackboneUGCableMaterial	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	LLQBranchManholes	Number LLQ Branch Manholes		= NumberLLQbranchcables * IF (LLQBranchBackboneUGDistance < 1, 0, LLQBranchBackboneUGDistance / VLOOKUP (Density, DistrSpacingTable, 2))	
AN	LLQStandardBranchManholeCost	LLQ Standard Branch Manhole Cost		= IF (OR (LLQBranchManholes = 0, LLQBranchDucts > 9) , 0, VLOOKUP (LLQBranchDucts, CHOOSE (FiberDepthCondition, ManholeHardRockTable, ManholeSoftRockTable, ManholeNormalSurfaceTable) , DensityColumnPointer))	
AO	LLQExpandedBranchManholeCost	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	LLQBackboneDucts	Number LLQ Backbone Ducts		= LLQMaxSizeBackboneCables + 2	
AQ	LLQBackboneManholes	Number LLQ Backbone Manholes		= IF (LLQBranchBackboneUGDistance < 1, 0, LLQBranchBackboneUGDistance / 2 / VLOOKUP (Density, DistrSpacingTable, 2))	

AR	LLQStandardBackboneManholeCost	LLQ Standard Backbone Manhole Cost		= IF (OR (LLQBackboneManholes = 0, LLQBackboneDucts > 9) , 0, VLOOKUP (LLQBackboneDucts, CHOOSE (FiberDepthCondition, ManholeHardRockTable, ManholeSoftRockTable, ManholeNormalSurfaceTable) , DensityColumnPointer))	
AS	LLQExpandedBackboneManholeCost	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	LLQUGStructureCost	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	PairsperURQbranchcable	Pairs Per URQ Branch Cable		= IF (NumberURQbranchcables = 0, 0, CEILING (NumberpairsinURquadrant / (2 * NumberURQbranchcables) / DistributionFillFactor, 1))	
AV	URQMaxSizeBranchCables	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	URQUGStructureCost	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)	

)ResidualBack :CableSize	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))	
)BranchBackb \erialDistance	LRQ Branch & Backbone Aerial Distance		= VLOOKUP (Density, CHOOSE (CopperDepthCondition, DistrHardMixTable, DistrSoftMixTable, DistrNormMixTable) , 4) * LRQBackboneBranchCableLength	
)BranchAerial eMaterial	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)))	
)BackboneAer ıbleMaterial	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)))	
)BranchBackb ;uriedDistanc	LRQ Branch & Backbone Buried Distance		= VLOOKUP (Density, CHOOSE (CopperDepthCondition, DistrHardMixTable, DistrSoftMixTable, DistrNormMixTable) , 3) * LRQBackboneBranchCableLength	
BranchBuried eMaterial	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)))	
BackboneBur ıbleMaterial	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)))	
BranchBackb GDistance	LRQ Branch & Backbone Underground Distance		= VLOOKUP (Density, CHOOSE (CopperDepthCondition, DistrHardMixTable, DistrSoftMixTable, DistrNormMixTable) , 2) * LRQBackboneBranchCableLength	
BranchUGCa aterial	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)))	
BackboneUG :Material	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)))	
BranchDucts	Number LRQ Branch Ducts		= LRQMaxSizeBranchCables + 2	
BranchManh	Number LRQ Branch Manholes		= NumberLRQbranchcables * IF (LRQBranchBackboneUGDistance < 1, 0, LRQBranchBackboneUGDistance / VLOOKUP (Density, DistrSpacingTable, 2))	
StandardBran nholeCost	LRQ Standard Branch Manhole Cost		= IF (OR (LRQBranchManholes = 0, LRQBranchDucts > 9) , 0, VLOOKUP (LRQBranchDucts, CHOOSE (FiberDepthCondition, ManholeHardRockTable, ManholeSoftRockTable, ManholeNormalSurfaceTable) , DensityColumnPointer))	
ExpandedBra anholeCost	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	
CX	PctDLCReqExRange	Percentage of DLC Lines Requiring Extended Range Line Cards		= IF (GridPlantType = "Cable", 0, ((PctULQReqExRange * (ULQHouseholds + ULQBusinessLines)) + (PctLLQReqExRange * (LLQHouseholds + LLQBusinessLines)) + (PctURQReqExRange * (URQHouseholds + URQBusinessLines)) + (PctLRQReqExRange * (LRQHouseholds + LRQBusinessLines))) / MAX (1, ((ULQHouseholds + ULQBusinessLines) + (LLQHouseholds + LLQBusinessLines) + (URQHouseholds + URQBusinessLines) + (LRQHouseholds + LRQBusinessLines))))	

CY	PctULQReqExRange	Percentage of ULQ Lines Requiring Extended Range Line Cards		= IF (ULQHorizontalfttoRoadCentroid + ULQverticalfttoRoadCentroid > BreakPointExRange, 1, IF (ULQHorizontalfttoRoadCentroid + ULQverticalfttoRoadCentroid + 1.5 * ULQBackboneandBranchCableLength < BreakPointExRange, 0, 1 - ((BreakPointExRange - (ULQHorizontalfttoRoadCentroid + ULQverticalfttoRoadCentroid)) / (1.5 * ULQBackboneandBranchCableLength))))	
CZ	PctLLQReqExRange	Percentage of LLQ Lines Requiring Extended Range Line Cards		= IF (LLQHorizontalfttoRoadCentroid + LLQverticalfttoRoadCentroid > BreakPointExRange, 1, IF (LLQHorizontalfttoRoadCentroid + LLQverticalfttoRoadCentroid + 1.5 * LLQBackboneBranchCableLength < BreakPointExRange, 0, 1 - ((BreakPointExRange - (LLQHorizontalfttoRoadCentroid + LLQverticalfttoRoadCentroid)) / (1.5 * LLQBackboneBranchCableLength))))	
DA	PctURQReqExRange	Percentage of URQ Lines Requiring Extended Range Line Cards		= IF (URQHorizontalfttoRoadCentroid + URQverticalfttoRoadCentroid > BreakPointExRange, 1, IF (URQHorizontalfttoRoadCentroid + URQverticalfttoRoadCentroid + 1.5 * URQBackboneBranchCableLength < BreakPointExRange, 0, 1 - ((BreakPointExRange - (URQHorizontalfttoRoadCentroid + URQverticalfttoRoadCentroid)) / (1.5 * URQBackboneBranchCableLength))))	
DB	PctLRQReqExRange	Percentage of LRQ Lines Requiring Extended Range Line Cards		= IF (LRQHorizontalfttoRoadCentroid + LRQverticalfttoRoadCentroid > BreakPointExRange, 1, IF (LRQHorizontalfttoRoadCentroid + LRQverticalfttoRoadCentroid + 1.5 * LRQBackboneBranchCableLength < BreakPointExRange, 0, 1 - ((BreakPointExRange - (LRQHorizontalfttoRoadCentroid + LRQverticalfttoRoadCentroid)) / (1.5 * LRQBackboneBranchCableLength))))	

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	NumberLLQRES Drops	Number Of LLQ Res Drops		= IF (TotalHousingUnitLocations = 0, 0, (LLQHousingUnitlocations / TotalHousingUnitLocations) * _1to2HUperDwellingGridLocations)	
W	AvgRESDropsper LLQterminal	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	TotalURQAerialD ropTerminalCost	Total URQ Aerial Drop Terminal Cost		= IF (URQDropTerminalPairs = 0, 0, URQTerminals * PercentDropsAerial * VLOOKUP (URQDropTerminalPairs, AerTermCostTable, DensityColumnPointer))	
AL	TotalURQBuried DropTerminalCost	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	URQAerialDropC ost	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	AvgRESDropsper LRQterminal	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	