AW	LLQRoadLength	LLQ Road Length		
AX	LLQArea	LLQ Road Reduced Area		
AY	LLQHorizontalftto RoadCentroid	LLQ Horizontal Feet To Road Centroid		
AZ	LLQverticalfttoRo adCentroid	LLQ Vertical Feet To Road Centroid		
BA	LRQHousingUnits	LRQ Housing Units		
BB	LRQHouseholds	LRQ Households		
BC	LRQBusinessLine s	LRQ Business Lines		
BD	LRQRoadLength	LRQ Road Length		
BE	LRQArea	LRQ Road Reduced Area		
BF	LRQHorizontalftto RoadCentroid	LRQ Horizontal Feet To Road Centroid		
BG	LRQverticalfttoRo adCentroid	LRQ Vertical Feet To Road Centroid		

Benchmark Cost Proxy Model Version 2.0 (BCPM2.0)
Copyright © 1997
Information contained in this document is proprietary to the BCPM2.0 Developers. The information may be used only in conjunction with the Benchmark Cost Proxy Model. Use for any other purpose is strictly prohibited.

Sheet: Output

Col	Range Name	Column Name	Column Comment	Formula
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 * TotalLinesServedInGrid
G		Lines Times Average Feeder Length		= AverageFeederDistance * TotalLinesServedInGrid
H		Area		= 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 Lines Served in Grid		= TotalLinesServedInGrid
O	AverageLoopLength	Average Loop Length		= IF (AverageFeederDistance = 10000, 0, AverageFeederDistance) + AverageDistributionDistance
P		Total Lines Loop Length		= TotalLinesServedInGrid * AverageLoopLength
Q		Total Distribution Cost		= TotalDistributionCost
R		Grand Total Feeder Cost		= GrandTotalFeederCost
S	UncappedTotalLoopCost	Uncapped Total Loop Cost		= TotalDistributionCost + GrandTotalFeederCost
T	ActualLoopCostPerLine	Actual Loop $ Per Line		= IF (TotalLinesServedInGrid = 0, 0, (UncappedTotalLoopCost / TotalLinesServedInGrid))
U	CappedLoopCostPerLine	Capped Loop $ Per Line		= IF (ActualLoopCostPerLine > InvestmentLoopCap, InvestmentLoopCap, ActualLoopCostPerLine)
V	CappedTotalLoopCost	Capped Total Loop Cost		= CappedLoopCostPerLine * TotalLinesServedInGrid
W		Land Investment		= SwitchingTotalInvestment * SwLandFactor

Benchmark Cost Proxy Model Version 2.0 (BCPM2.0)
Copyright © 1997
Information contained in this document is proprietary to the BCPM2.0 Developers. The information may be used only in conjunction with the Benchmark Cost Proxy Model. Use for any other purpose is strictly prohibited.

X		Building Investment		= SwitchingTotalInvestment * SwBldgFactor
Y		Central Office Switching Investment		= SwitchingTotalInvestment
Z	TotalDLCDS1Investment	Total DLC, DS1 Electronic Investment		= TotalElectronics
AA	TotalInterOfficeFacilities	Interoffice Facilities		= SwitchingTotalInvestment * InterofficeSwRatio
AB		Uncapped Total Circuit		= TotalDLCDS1Investment + TotalInterOfficeFacilities
AC		Uncapped NID Cost		= TotalGridNIDCost
AD		Uncapped Aerial Copper Cable Investment		= AerialCopperCableInvestment
AE		Uncapped Underground Copper Cable Investment		= UndergroundCopperCableInvestment
AF		Uncapped Buried Copper Cable Investment		= BuriedCopperCableInvestment
AG		Uncapped Aerial Fiber Cable Investment		= AerialFiberCableInvestment
AH		Uncapped Underground Fiber Cable Investment		= UndergroundFiberCableInvestment
AI		Uncapped Buried Fiber Cable Investment		= BuriedFiberCableInvestment
AJ		Uncapped Pole Line Investment		= PoleInvestmentAssignedCopper + PoleInvestmentAssignedFiber
AK		Uncapped Conduit Investment		= ConduitInvestmentAssignedCopper + ConduitInvestmentAssignedFiber

Benchmark Cost Proxy Model Version 2.0 (BCPM2.0)
Copyright © 1997
Information contained in this document is proprietary to the BCPM2.0 Developers. The information may be used only in conjunction with the Benchmark Cost Proxy Model. Use for any other purpose is strictly prohibited.

Sheet: Grid Demographics

Col	Range Name	Column Name	Column Comment	Formula
A	CLLIQuad*	CLLIQUAD		= CLLI&LEFT (FDICode, 1)
B	CLLISegment*	CLLISEGMENT		= CLLI&LEFT (FDICode, 2)
C	StateId	State Id		= MID (CLLI, 5, 2)
D	CompanySize	Company Size	Company Size: 1 = Small 2 = Medium 3 = Large	= VLOOKUP (CLLI,CompanySizeTable, 2)
E	FeederSegmentSequenceNumber*	Feeder Segment Sequence Number		= COUNTIF (INDIRECT (ADDRESS (ROW (AbsoluteCLLIQuad) , COLUMN (AbsoluteCLLIQuad)) &":"&ADDRESS (ROW (CLLIQuad) , COLUMN (CLLIQuad))) , CLLIQuad)
F	SingleResidenceLines	Single Residence Lines		= TotalHouseholds
G	AdditionalResidenceLines	Additional Residence Lines		= TotalHouseholds * (VLOOKUP (StateId, StateInformationTable, 2) - 1)
H	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)))
I	ResidenceLines	Total Residence Lines		= ResidenceLineBalancingFctr * (SingleResidenceLines + AdditionalResidenceLines)
J	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)))
K	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)))
L	BusinessLineBalancingFctr	Business Line Balancing Factor		= IF (TotalBusinessLines = 0, 0, (SingleBusinessLines + MultiBusinessLines) / TotalBusinessLines)
M	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)))
N	TotalLinesServedInGrid	Total Lines Served In Grid		= ResidenceLines + SingleBusinessLines + MultiBusinessLines + SpecialAccessLines
O	Density*	Density in Lines/SQ MI		= IF (Area_sq_Miles = 0, IF (ISNUMBER (PrevDensity) , PrevDensity, NextDensity) , TotalLinesServedInGrid / Area_sq_Miles)
P	DensityColumnPointer	Density Column Pointer		= VLOOKUP (Density, DensityPointerTable, 2)
Q	GridLinesProvisionedAsDS1s	Grid Lines Provisioned as DS-1s		= SpecialAccessLines * VLOOKUP (TotalLinesServedInGrid, VoiceGradeRatioTable, 5) + SwitchedLinesInGrid * VLOOKUP (TotalLinesServedInGrid, VoiceGradeRatioTable, 3)
R	SwitchedLinesInGrid	Switched Lines in Grid		= TotalLinesServedInGrid - SpecialAccessLines

Benchmark Cost Proxy Model Version 2.0 (BCPM2.0)
Copyright © 1997
Information contained in this document is proprietary to the BCPM2.0 Developers. The information may be used only in conjunction with the Benchmark Cost Proxy Model. Use for any other purpose is strictly prohibited.

S	SoilSlopeFactor	Soil Slope Factor		= IF (MinimumSoilSlope > MinSlopeTrigger, IF (MaximumSoilSlope > MaxSlopeTrigger, CombSlopeFactor, MinSlopeFactor) , IF (MaximumSoilSlope > MaxSlopeTrigger, MaxSlopeFactor, 1))
T	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))
U	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))
V	WaterCostMultiplier	Water Cost Multiplier		= IF (WaterTableDepth < CriticalWaterDepth, 1 + (WaterFactor / 100) , 1)
W	SurfaceIndicator	Surface Indicator		= IF (ISBLANK (SurfaceSoilTexture) , 0, IF (ISERROR (VLOOKUP (SurfaceSoilTexture, SurfaceTextureTable, 2)) = TRUE, 0, VLOOKUP (SurfaceSoilTexture, SurfaceTextureTable, 2)))
X	MainFeederDistance*	Main Feeder Distance		= MainFeederLength
Y	PrimarySubfeederDistance	Primary Subfeeder Distance		= SubfeederLength
Z	SubfeederPart2Distance	Subfeeder Part 2 Distance		= Part2SubfeederLength
AA	MaxDLCTOFDICableDistance	Maximum DLC to FDI Cable Distance		= MAX (ULQHorizontalfttoRoadCentroid + ULQverticalfttoRoadCentroid, LLQHorizontalfttoRoadCentroid + LLQVerticalFtToRoadCentroid, URQHorizontalfttoRoadCentroid + URQverticalfttoRoadCentroid, LRQHorizontalfttoRoadCentroid + LRQVerticalFtToRoadCentroid)
AB	TotalFeederPlanningDistance	Total Feeder Planning Distance		= SoilSlopeFactor * (MainFeederDistance + PrimarySubfeederDistance + SubfeederPart2Distance + MaxDLCToFDICableDistance)
AC	GridSideLength	Grid Side Length		= (SQRT (CHOOSE (LargestQuadAreaNumber, ULQArea, LLQArea, LRQArea, URQArea)) * 5280) * SoilSlopeFactor
AD	NumberOfLotsInGrid	Number of lots in Grid		= CHOOSE (LargestQuadAreaNumber, ULQHousingUnits + ULQBusinesslocations, LLQHousingUnits + LLQBusinesslocations, LRQHousingUnits + LRQBusinesslocations, URQHousingUnits + URQBusinesslocations)
AE	NumberOfLotsPerBaseSide	Number of Lots per Base Side		= MAX (2, CEILING (SQRT (NumberOfLotsInGrid) , 1))
AF	BaseSideLotLength	Base Side Lot Length		= GridSideLength / NumberOfLotsPerBaseSide
AG	SecondSideLotLength	Second Side Lot Length		= IF (SQRT (NumberOfLotsInGrid) = 0, 0, BaseSideLotLength * CEILING (SQRT (NumberOfLotsInGrid) , 1) / SQRT (NumberOfLotsInGrid))
AH	MaxDistributionPlanningDistance	Maximum Distribution Planning Distance		= SoilSlopeFactor * (GridSideLength - BaseSideLotLength - SecondSideLotLength)
AI	TotalLoopPlanningLength	Total Loop Planning Length		= TotalFeederPlanningDistance + MaxDistributionPlanningDistance
AJ	FeederFill	Feeder Fill Factor		= VLOOKUP (Density, DensityFillTable, 2)
AK	DistributionFillFactor	Distribution Fill Factor		= VLOOKUP (Density, DensityFillTable, 3)

Benchmark Cost Proxy Model Version 2.0 (BCPM2.0)
Copyright © 1997
Information contained in this document is proprietary to the BCPM2.0 Developers. The information may be used only in conjunction with the Benchmark Cost Proxy Model. Use for any other purpose is strictly prohibited.

AL	GridPlantType*	Grid Plant Type		= IF (AND (TotalLinesServedInGrid = 0, FdrBranch0) , NextGridplanttype, IF (AND (TotalLoopPlanningLength < Breakpoint, (NetCopperPairs / FeederFill) < MaxDistrSize) , "Cable", IF (AND (TotalLoopPlanningLength > = Breakpoint, TotalLinesServedInGrid / ElectronicFill < = 240) , "DLC - S", "DLC - L")))
AM	GridPlantIndex	Grid Plant Index		= IF (GridPlantType = "Cable", 1, IF (GridPlantType = "DLC - S", 2, 3))
AN	IsSamePlantType	Same Plant Type		= GridPlantType = NextGridplanttype
AO	NetCopperPairs	Net Grid Pairs if copper		= TotalLinesServedInGrid - GridLinesProvisionedAsDS1s + GridLinesProvisionedAsDS1s / 12
AP	VoiceGradeLinesEquippedatTerminal	# Voice Grade Lines Equipped at Terminal		= IF (TotalLinesServedInGrid = 0, 0, CHOOSE (GridPlantIndex, 0, CEILING (TotalLinesServedInGrid / ElectronicFill, 6) , CEILING ((SwitchedLinesInGrid * VLOOKUP (TotalLinesServedInGrid, VoiceGradeRatioTable, 2) + SpecialAccessLines * VLOOKUP (TotalLinesServedInGrid, VoiceGradeRatioTable, 4)) / ElectronicFill, 4)))
AQ	NumberDLCLsatFullCapacity	# DLC-L Systems at Full Capacity		= CHOOSE (GridPlantIndex, 0, 0, TRUNC (VoiceGradeLinesEquippedAtTerminal / 1344))
AR	SwitchedLinesinOverflowDLCLTerminal	# Switched Lines in Overflow DLC-L Terminal		= CHOOSE (GridPlantIndex, 0, 0, VoiceGradeLinesEquippedAtTerminal - (1344 * NumberDLCLsAtFullCapacity))
AS	LinesatDS1atTerminal	# DS-1 Lines at Terminal		= IF (TotalLinesServedInGrid = 0, 0, CHOOSE (GridPlantIndex, 0, 0, GridLinesProvisionedAsDS1s / HiCapFill))
AT	DS3DS1UnitsperTerminalLocation	# DS-3/DS-1 Units per Terminal Location		= CEILING (LinesatDS1AtTerminal / (672 * ElectronicFill) , 1)
AU	NumberFibersPerGrid	Number Fibers per Grid		= CHOOSE (GridPlantIndex, 0, (1 + DS3DS1UnitsPerTerminalLocation) * 4, (NumberDLCLsAtFullCapacity + 1 + DS3DS1UnitsPerTerminalLocation) * 4)
AV	DLCSFibersThisGrid	# DLC-S Fibers this Grid		= CHOOSE (GridPlantIndex, 0, NumberFibersPerGrid, 0)
AW	DLCLFibersThisGrid	# DLC-L Fibers this Grid		= CHOOSE (GridPlantIndex, 0, 0, NumberFibersPerGrid)
AX	AverageDLCtoFDIDistance	Average DLC to FDI Distance		= AVERAGE (ULQHorizontalfttoRoadCentroid + ULQverticalfttoRoadCentroid, LLQHorizontalfttoRoadCentroid + LLQVerticalFtToRoadCentroid, URQHorizontalfttoRoadCentroid + URQverticalfttoRoadCentroid, LRQHorizontalfttoRoadCentroid + LRQVerticalFtToRoadCentroid)
AY	AverageFeederDistance	Average Feeder Distance		= MainFeederLength + SubfeederLength + SubfeederPart2Distance + AverageDLCToFDIDistance
AZ	FdrBranch0	Is Feeder Branch 0		= MID (FDICode, 2, 1) = "0"
BA	FeederBranchNumber*	Feeder Branch Number		= LEFT (FDICode, 2)
BB	MainFeederSequenceNumber*	Main Feeder Sequence Number		= COUNTIF (INDIRECT (ADDRESS (ROW (GridDemographicsFirstRow) , COLUMN (FeederBranchNumber)) &":"&ADDRESS (ROW (FeederBranchNumber) , COLUMN (FeederBranchNumber))) , FeederBranchNumber)
BC	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)))
BD	_1to2Buildings	2 HU Per Buildings		= _2HUPerDwellinglocations / (VLOOKUP (2, HousingUnitsperDwelling, DensityColumnPointer, FALSE))
BE	_3to4Buildings	3-4 HU Per Buildings		= _3to4HUperdwellinglocations / (VLOOKUP ("3 - 4", HousingUnitsperDwelling, DensityColumnPointer, FALSE))

Benchmark Cost Proxy Model Version 2.0 (BCPM2.0)
Copyright © 1997
Information contained in this document is proprietary to the BCPM2.0 Developers. The information may be used only in conjunction with the Benchmark Cost Proxy Model. Use for any other purpose is strictly prohibited.

BF	_5to9Buildings	5-9 HU Per Buildings		= _5to9HUperdwellinglocations / (VLOOKUP ("5 - 9", HousingUnitsperDwelling, DensityColumnPointer, FALSE))
BG	_10to19Buildings	10-19 HU Per Buildings		= _10to25HUperdwellinglocations / (VLOOKUP ("10 - 19", HousingUnitsperDwelling, DensityColumnPointer, FALSE))
BH	_20-49Buildings	20-49 HU Per Buildings		= _25to50HUperdwellinglocations / (VLOOKUP ("20 - 49", HousingUnitsperDwelling, DensityColumnPointer, FALSE))
BI	Over50Buildings	>50 HU Per Buildings		= Over50HUPerDwellingLocations / (VLOOKUP (" > 50", HousingUnitsperDwelling, DensityColumnPointer, FALSE))
BJ	OtherBuildings	Other HU Buildings		= OtherHULocations / (VLOOKUP ("Other", HousingUnitsperDwelling, DensityColumnPointer, FALSE))

Benchmark Cost Proxy Model Version 2.0 (BCPM2.0)
Copyright © 1997
Information contained in this document is proprietary to the BCPM2.0 Developers. The information may be used only in conjunction with the Benchmark Cost Proxy Model. Use for any other purpose is strictly prohibited.

Sheet: Switching

Col	Range Name	Column Name	Column Comment	Formula
A	SwitchingCLLI	CLLI		= CLLI
B	SwitchingSwitchedLinesEquipped	Switched Lines Equipped		= SwitchedLinesInGrid
C	TotalLinesEquippedInSwitch	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 * SwitchFillFactor * (1 + VLOOKUP (CompanySize, COSwitchCostTable, 6)))
E	SwitchingVariableCostPerLine	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

Benchmark Cost Proxy Model Version 2.0 (BCPM2.0)
Copyright © 1997
Information contained in this document is proprietary to the BCPM2.0 Developers. The information may be used only in conjunction with the Benchmark Cost Proxy Model. Use for any other purpose is strictly prohibited.

Sheet: Main Feeder

Col	Range Name	Column Name	Column Comment	Formula
A	MainFeederQuadrant	Quadrant Number		= LEFT (FDICode, 1)
B	MainFeederBranch*	Main Feeder Branch Number		= MID (FDICode, 2, 1)
C		Feeder Sequence Number	Data Source: Grid Demographics	= MainFeederSequenceNumber
D	MainFdrSegment Distance	Main Feeder Segment Distance		= IF (IsFirstSegmentInBranch, MainFeederDistance, IF (IsSameAsPrevFeederBranch, MainFeederDistance - PrevMainFeederDistance, MainFeederDistance - 10000))
E	SegmentType1*	Segment Type 1		= CHOOSE (GridPlantIndex, "Cable", "Fiber", "Fiber")
F	CableTestColumn	Same Quadrant And Cable		= IF (AND (OR (IsSameAsNextCLLIQuad, MainFeederFirstRow = "1") , SegmentType1 = "Cable") , CLLIQuad, "")
G	FiberBranchTestColumn	Same Branch And Cable		= IF (SegmentType1 = "Cable", IF (MainFeederBranch = "0", "", IF (MainFeederBranch = "1", CLLIQuad&"2", CLLIQuad&"1")) , "")
H	FiberTestColumn	Same Quadrant And Fiber		= IF (QuadrantContainFiber, CLLIQuad, "")
I	FiberBranchTestColumn	Same Branch And Fiber		= IF (QuadrantContainFiber, IF (MainFeederBranch = "0", "", IF (MainFeederBranch = "1", CLLIQuad&"2", CLLIQuad&"1")) , "")
J	WorkingPrsfromSubfeeders	Working Pairs from Subfeeders		= IF (IsFirstSubfeederSegment, IF (SlopeAdjSubfeederDistance = 0, NetCopperPairs, CumulativeCopperSubfeederPairs * FeederFill) , 0)
K	MainFdrCopperWrkgLines	Main Feeder Copper Working Lines		= SUMIF (OFFSET (CableTestColumnRelative, 0, 0, LastDataRow, 1) , CLLIQuad, OFFSET ($J3, 0, 0, LastDataRow, 1)) - SUMIF (OFFSET (CableBranchTestColumnRelative, 0, 0, LastDataRow, 1) , CLLISegment, OFFSET ($J3, 0, 0, LastDataRow, 1))
L	WorkingFibersfromSubfeeders	Working Fibers From Subfeeders		= IF (IsFirstSubfeederSegment, IF (SlopeAdjSubfeederDistance = 0, NumberFibersPerGrid, NumberSubfeederFibers) , 0)
M	MainFdrWrkgfibers	Main Feeder Working fibers		= SUMIF (OFFSET (FiberTestColumnRelative, 0, 0, LastDataRow, 1) , CLLIQuad, OFFSET ($L3, 0, 0, LastDataRow, 1)) - SUMIF (OFFSET (FiberBranchTestColumnRelative, 0, 0, LastDataRow, 1) , CLLISegment, OFFSET ($L3, 0, 0, LastDataRow, 1))
N	MainFdrFibersNeeded	Main Feeder Fibers Needed		= MainFdrWrkgFibers / FeederFill
O	CopperMainFdrPrsNeeded	Copper Main Feeder Pairs Needed		= MainFdrCopperWrkgLines / FeederFill
P	NumberMaxSize MainFdrCopperCables	Number of Max Size Main Feeder Copper Cables		= TRUNC (IF (CopperMainFdrPrsNeeded > MaxFeederSize, CopperMainFdrPrsNeeded / MaxFeederSize, 0))
Q	ResidualCopperMainFdrSize	Residual Copper Main Feeder Size		= IF (CopperMainFdrPrsNeeded > 0, INDEX (FeederCableSize, MATCH (CopperMainFdrPrsNeeded - (MaxFeederSize * NumberMaxSizeMainFdrCopperCables) , FeederCableSize, - 1) , 1) , 0)

Benchmark Cost Proxy Model Version 2.0 (BCPM2.0)
Copyright © 1997
Information contained in this document is proprietary to the BCPM2.0 Developers. The information may be used only in conjunction with the Benchmark Cost Proxy Model. Use for any other purpose is strictly prohibited.

R	CopperMainFdrStructureRatio	Copper Main Feeder Structure %		= IF (CopperMainFdrPrsNeeded < 1, 0, IF (NumberMaxSizeMainFdrCopperCables > 0, Over4200, IF (MainFdrFibersNeeded > 0, VLOOKUP (ResidualCopperMainFdrSize, StructureAllocationTable, 2) , 1)))
S	FiberMainFdrStructureRatio	Fiber Main Feeder Structure %		= 1 - CopperMainFdrStructureRatio
T	NumberMaxSizeMainFdrFiberCables	Number Max Size Main Feeder Fiber Cables		= TRUNC (IF (MainFdrFibersNeeded > MaxFiberSize, MainFdrFibersNeeded / MaxFiberSize, 0))
U	ResidualFiberMainFdrSize	Residual Fiber Main Feeder Size		= IF (MainFdrFibersNeeded = 0, 0, INDEX (FiberCableSize, MATCH (MainFdrFibersNeeded - (MaxFiberSize * NumberMaxSizeMainFdrFiberCables) , FiberCableSize, - 1) , 1))
V	IsMainFeederCopperOnly	Main Feeder Copper Only?		= ResidualFiberMainFdrSize < = 0
W	MainFdrCopperUGDistance	Main Feeder Copper Underground Distance		= IF (IsMainFeederCopperOnly, VLOOKUP (Density,CopperPlantMixTable, 2) , VLOOKUP (Density, FiberPlantMixTable, 2)) * MainFdrSegmentDistance * SoilSlopeFactor
X	MainFdrSegmentUGDistance	Main Feeder Segment Underground Copper Material $		= IF (ResidualCopperMainFdrSize = 0, 0, MainFdrSegmentUGDistance * (VLOOKUP (ResidualCopperMainFdrSize, IF (CopperGauge = 26, UGCopper26Table, UGCopper24Table) , DensityColumnPointer, FALSE) + (NumberMaxSizeMainFdrFiberCables * VLOOKUP (MaxFeederSize, IF (CopperGauge = 26, UGCopper26Table, UGCopper24Table) , DensityColumnPointer, FALSE)))) * CopperCostRatio
Y	MainFdrCopperBurDistance	Main Feeder Copper Buried Distance		= IF (IsMainFeederCopperOnly, VLOOKUP (Density,CopperPlantMixTable, 3) , VLOOKUP (Density, FiberPlantMixTable, 3)) * MainFdrSegmentDistance * SoilSlopeFactor
Z	MainFdrSegmentBurCopperMaterial	Main Feeder Segment Buried Copper Material $		= IF (ResidualCopperMainFdrSize = 0, 0, MainFdrCopperBurDistance * (VLOOKUP (ResidualCopperMainFdrSize, IF (CopperGauge = 26, BURCopper26Table, BURCopper24Table) , DensityColumnPointer, FALSE) + (NumberMaxSizeMainFdrCopperCables * VLOOKUP (MaxFeederSize, IF (CopperGauge = 26, BURCopper26Table, BURCopper24Table) , DensityColumnPointer, FALSE)))) * CopperCostRatio
AA	MainFdrCopperAerDistance	Main Feeder Copper Aerial Distance		= IF (IsMainFeederCopperOnly, VLOOKUP (Density,CopperPlantMixTable, 4) , VLOOKUP (Density, FiberPlantMixTable, 4)) * MainFdrSegmentDistance * SoilSlopeFactor
AB	MainFdrSegmentAerDistance	Main Feeder Segment Aerial Copper Material $		= IF (ResidualCopperMainFdrSize = 0, 0, MainFdrSegmentAerDistance * (VLOOKUP (ResidualCopperMainFdrSize, IF (CopperGauge = 26, AERCopper26Table, AERCopper24Table) , DensityColumnPointer, FALSE) + (NumberMaxSizeMainFdrCopperCables * VLOOKUP (MaxFeederSize, IF (CopperGauge = 26, AERCopper26Table, AERCopper24Table) , DensityColumnPointer, FALSE)))) * CopperCostRatio
AC	MainFdrSegmentAerCopperMaterial	Total Main Feeder Copper Material $		= MainFdrSegmentUGCopperMaterial + MainFdrSegmentBurCopperMaterial + MainFdrSegmentAerCopperMaterial
AD	MainFdrSegmentUGFiberDistance	Main Feeder Segment Underground Fiber Distance		= IF (IsMainFeederCopperOnly, 0, MainFdrSegmentUGDistance)

Benchmark Cost Proxy Model Version 2.0 (BCPM2.0)
Copyright © 1997
Information contained in this document is proprietary to the BCPM2.0 Developers. The information may be used only in conjunction with the Benchmark Cost Proxy Model. Use for any other purpose is strictly prohibited.

AE	MainFdrSegment UGFiberMaterial	Main Feeder Segment Underground Fiber Material $		= IF (MainFdrSegmentUGFiberDistance = 0, 0, MainFdrSegmentUGFiberDistance * (VLOOKUP (ResidualFiberMainFdrSize, UGFiberTable, DensityColumnPointer, FALSE) + (NumberMaxSizeMainFdrFiberCables * VLOOKUP (MaxFiberSize, UGFiberTable, DensityColumnPointer)))) * FiberCostRatio
AF	MainFdrSegment BurFiberDistance	Main Feeder Segment Buried Fiber Distance		= IF (IsMainFeederCopperOnly, 0, MainFdrCopperBurDistance)
AG	MainFdrSegment BurFiberMaterial	Main Feeder Segment Buried Fiber Material $		= IF (MainFdrSegmentBurFiberDistance = 0, 0, MainFdrSegmentBurFiberDistance * (VLOOKUP (ResidualFiberMainFdrSize, BURFiberTable, DensityColumnPointer, FALSE) + (NumberMaxSizeMainFdrFiberCables * VLOOKUP (MaxFiberSize, BURFiberTable, DensityColumnPointer)))) * FiberCostRatio
AH	MainFdrSegment AerFiberDistance	Main Feeder Segment Aerial Fiber Distance		= IF (IsMainFeederCopperOnly, 0, MainFdrSegmentAerDistance)
AI	MainFdrSegment AerFiberMaterial	Main Feeder Segment Aerial Fiber Material $		= IF (MainFdrSegmentAerFiberDistance = 0, 0, MainFdrSegmentAerFiberDistance * (VLOOKUP (ResidualFiberMainFdrSize, AERFiberTable, DensityColumnPointer, FALSE) + (NumberMaxSizeMainFdrFiberCables * VLOOKUP (MaxFiberSize, AERFiberTable, DensityColumnPointer)))) * FiberCostRatio
AJ	TotalMainFdrFiberMaterial	Total Main Feeder Fiber Material $		= MainFdrSegmentUGFiberMaterial + MainFdrSegmentBurFiberMaterial + MainFdrSegmentAerFiberMaterial
AK	NumberMainFeederDucts	Number Main Feeder Ducts		= IF (MainFdrSegmentUGDistance < 1, 0, IF (ResidualCopperMainFdrSize > 0, 1, 0) + NumberMaxSizeMainFdrCopperCables + IF (ResidualFiberMainFdrSize > 0, 1, 0) + IF (AirlineDistance = "", 0, 1))
AL	NumberMainFdrManholes	Number Main Feeder Manholes		= IF (NumberMainFeederDucts < 1, 0, CEILING (MainFdrSegmentUGDistance / VLOOKUP (Density, FeederSpacingTable, 2) , 1) + IF (AirlineDistance = "", 0, IF (FeederSegmentSequenceNumber = 1, 1, 0)))
AM	MainFeederStandardManholeCost	Main Feeder Standard Manhole Cost		= IF (NumberMainFdrManholes = 0, 0, IF (NumberMainFeederDucts < = 9, VLOOKUP (NumberMainFeederDucts, CHOOSE (FiberDepthCondition, ManholeHardRockTable, ManholeSoftRockTable, ManholeNormalSurfaceTable) , DensityColumnPointer) , 0))
AN	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))
AO	NumberMainFdrPoles	Number Main Feeder Poles		= IF (MainFdrSegmentAerDistance < 1, 0, CEILING (MainFdrSegmentAerDistance / VLOOKUP (Density, FeederSpacingTable, 3) , 1) + IF (AirlineDistance = "", 0, 1))
AP	MainFeederUGCopperStructure	Main Feeder Underground Copper Structure $		= IF (NumberMainFeederDucts = 0, 0, CopperMainFdrStructureRatio * WaterCostMultiplier * (IF (NumberMainFeederDucts < = 9, MainFeederStandardManholeCost, MainFeederExpandedManholeCost) * NumberMainFdrManholes + IF (MainFdrSegmentUGFiberDistance > = MainFdrCopperUGDistance, MainFdrSegmentUGFiberDistance, MainFdrCopperUGDistance) * (NumberMainFeederDucts * VLOOKUP (0, ConduitDensityTable, DensityColumnPointer) + VLOOKUP (Density, CHOOSE (FiberDepthCondition, HardRockStructure, SoftRockStructure, NormalStructure) , 2))))
AQ	MainFeederBuriedCopperStructure	Main Feeder Buried Copper Structure $		= IF (MainFdrCopperBurDistance + MainFdrSegmentBurFiberDistance = 0, 0, CopperMainFdrStructureRatio * WaterCostMultiplier * IF (MainFdrSegmentBurFiberDistance > = MainFdrCopperBurDistance, MainFdrSegmentBurFiberDistance, MainFdrCopperBurDistance) * IF (MainFdrSegmentBurFiberDistance = 0, VLOOKUP (Density, CHOOSE (CopperDepthCondition, HardRockStructure, SoftRockStructure, NormalStructure) , 4) , VLOOKUP (Density, CHOOSE (FiberDepthCondition, HardRockStructure, SoftRockStructure, NormalStructure) , 4)))

Benchmark Cost Proxy Model Version 2.0 (BCPM2.0)
Copyright © 1997
Information contained in this document is proprietary to the BCPM2.0 Developers. The information may be used only in conjunction with the Benchmark Cost Proxy Model. Use for any other purpose is strictly prohibited.

AR	MainFeederAerialCopperStructure	Main Feeder Aerial Copper Structure $		= IF (NumberMainFdrPoles = 0, 0, CopperMainFdrStructureRatio * WaterCostMultiplier * NumberMainFdrPoles * VLOOKUP (Density, CHOOSE (FiberDepthCondition, HardRockStructure, SoftRockStructure, NormalStructure) , 6))
AS	TotalMainFdrCopperStructure	Total Main Feeder Copper Structure $		= MainFeederUGCopperStructure + MainFeederBuriedCopperStructure + MainFeederAerialCopperStructure
AT	TotalMainFdrCopperCost	Total Main Feeder Copper Cost		= TotalMainFdrCopperMaterial + TotalMainFdrCopperStructure
AU	CopperMainFeederCostPerLine	Copper Main Feeder Cost Per Line		= IF (CopperMainFdrPrsNeeded = 0, 0, TotalMainFdrCopperCost / CopperMainFdrPrsNeeded)
AV	CumulativeMainFdrCopperCostperPair	Cumulative Main Feeder Copper Cost per Pair		= SUMIF (OFFSET (CableTestColumn, 0, 0, ROW () - ROW (MainFeederFirstRow) + 1, 1) , CLLIQuad, OFFSET (CopperMainFeederCostPerLine, 0, 0, ROW () - ROW (MainFeederFirstRow) + 1, 1)) - SUMIF (OFFSET (CableBranchTestColumn, 0, 0, ROW () - ROW (MainFeederFirstRow) + 1, 1) , CLLISegment, OFFSET (CopperMainFeederCostPerLine, 0, 0, ROW () - ROW (MainFeederFirstRow) + 1, 1))
AW	MainFeederUGFiberStructure	Main Feeder Underground Fiber Structure $		= IF (NumberMainFeederDucts = 0, 0, FiberMainFdrStructureRatio * WaterCostMultiplier * (IF (NumberMainFeederDucts < = 9, MainFeederStandardManholeCost, MainFeederExpandedManholeCost) * NumberMainFdrManholes + MainFdrSegmentUGFiberDistance * (NumberMainFeederDucts * VLOOKUP (0, ConduitDensityTable, DensityColumnPointer) + VLOOKUP (Density, CHOOSE (FiberDepthCondition, HardRockStructure, SoftRockStructure, NormalStructure) , 2))))
AX	MainFeederBuriedFiberStructure	Main Feeder Buried Fiber Structure $		= IF (MainFdrSegmentBurFiberDistance = 0, 0, FiberMainFdrStructureRatio * WaterCostMultiplier * MainFdrSegmentBurFiberDistance * VLOOKUP (Density, CHOOSE (FiberDepthCondition, HardRockStructure, SoftRockStructure, NormalStructure) , 4))
AY	MainFeederAerialFiberStructure	Main Feeder Aerial Fiber Structure $		= IF (NumberMainFdrPoles = 0, 0, FiberMainFdrStructureRatio * WaterCostMultiplier * NumberMainFdrPoles * VLOOKUP (Density, CHOOSE (FiberDepthCondition, HardRockStructure, SoftRockStructure, NormalStructure) , 6))
AZ	TotalMainFdrFiberStructure	Total Main Feeder Fiber Structure $		= MainFeederUGFiberStructure + MainFeederBuriedFiberStructure + MainFeederAerialFiberStructure
BA	TotalMainFdrFiberCost	Total Main Feeder Fiber Cost		= TotalMainFdrFiberMaterial + TotalMainFdrFiberStructure
BB	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 will work*
BC	CumulativeMainFdrFiberCostperFiber	Cumulative Main Feeder Fiber Cost per Fiber		= SUMIF (OFFSET (FiberTestColumn, 0, 0, ROW () - ROW (MainFeederFirstRow) + 1, 1) , CLLIQuad, OFFSET (MainFeederFiberCostPerFiber, 0, 0, ROW () - ROW (MainFeederFirstRow) + 1, 1)) - SUMIF (OFFSET (FiberBranchTestColumn, 0, 0, ROW () - ROW (MainFeederFirstRow) + 1, 1) , CLLISegment, OFFSET (MainFeederFiberCostPerFiber, 0, 0, ROW () - ROW (MainFeederFirstRow) + 1, 1))
BD	IsSameAsPrevFeederBranch	Same As Previous Feeder Branch		= AND (IsSameAsPrevCLLIQuad, MainFeederBranch = PrevMainFeederBranch)
BE	IsSameAsPrevCLLIQuad	Same As Previous CLLI Quad		= (CLLIQuad = PrevCLLIQuad)
BF	IsSameAsNextCLLIQuad	Same As Next CLLI Quad		= (CLLIQuad = NextCLLIQuad)

Benchmark Cost Proxy Model Version 2.0 (BCPM2.0)
Copyright © 1997
Information contained in this document is proprietary to the BCPM2.0 Developers. The information may be used only in conjunction with the Benchmark Cost Proxy Model. Use for any other purpose is strictly prohibited.

BG	IsLastCLLIRecord	Last Record In CLLI		= OR (CLLI < > NextCLLI, NextCLLIQuad = "", ISBLANK (NextCLLIQuad))
BH	IsFirstSegmentInBranch	First Segment In Branch		= NOT (IsSameAsPrevCLLIQuad)
BI	QuadrantContainFiber*	Quadrant Contain Fiber		= IF (SegmentType1 = "Fiber", TRUE, (AND (IsSameAsNextCLLIQuad, NextQuadrantContainFiber = TRUE)))

Benchmark Cost Proxy Model Version 2.0 (BCPM2.0)
Copyright © 1997
Information contained in this document is proprietary to the BCPM2.0 Developers. The information may be used only in conjunction with the Benchmark Cost Proxy Model. Use for any other purpose is strictly prohibited.

Sheet: SubFeeder

Col	Range Name	Column Name	Column Comment	Formula
A	SubfeederNumber *	Subfeeder Number		= LEFT (FDICode, 5)
B	SubfeederSegment Number	Subfeeder Segment Number		= COUNTIF (INDIRECT (ADDRESS (ROW (SubfeederFirstRow) , COLUMN (SubfeederFirstRow)) &":"&ADDRESS (ROW (SubfeederNumber) , COLUMN (SubfeederNumber))) , SubfeederNumber)
C	IsFirstSubfeederSegement	First Subfeeder Segment		= SubfeederSegmentNumber = 1
D	IsLastCLLIRecord	Last CLLI Record		= AND (CLLIQuad < > NextCLLIQuad, NextCLLIQuad = "", ISBLANK (NextCLLIQuad))
E	IsSameAsNextQuadrant	Same As Next Quadrant		= (CLLIQuad = NextCLLIQuad)
F	IsSameAsNextSubfeeder	Same As Next Subfeeder		= (SubfeederNumber = NextSubfeederNumber)
G	IsSameAsNextCLLI	Same As Next CLLI		= (CLLI = NextCLLI)
H	SlopeAdjSubfeederDistance	Slope Adjusted Subfeeder Distance		= PrimarySubfeederDistance * SoilSlopeFactor
I		Grid Plant Type		= GridPlantType
J		Number Of Fibers From This Grid		= NumberFibersPerGrid
K	CumulativeDLCSLinesperSubfeeder *	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	NumberSubfeederFibers*	Number Subfeeder Fibers This segment		= IF (IsSameAsNextSubfeeder, IF (AND (CumulativeDLCSLinesperSubfeeder > 0, CumulativeDLCSLinesperSubfeeder < 240 * ElectronicFill, NumberFibersPerGrid = 4, NextNumberSubfeederFibers = 4) , 4, NumberFibersPerGrid + NextNumberSubfeederFibers) , NumberFibersPerGrid)
M	SubfeederFiberSize	Subfeeder Fiber Size		= IF (OR (SlopeAdjSubfeederDistance = 0, GridPlantType = "Cable") , 0, INDEX (FiberCableSize, MATCH (NumberSubfeederFibers, FiberCableSize, - 1) , 1))
N	SubfeederCopperPairsthisGrid	Subfeeder Copper Pairs This Grid		= CHOOSE (GridPlantIndex, NetCopperPairs / FeederFill, 0, 0)
O	CumulativeCopperSubfeederPairs*	Cumulative Copper Subfeeder Pairs		= IF (IsSameAsNextSubfeeder, SubfeederCopperPairsthisGrid + NextCumulativeCopperSubfeederPairs, SubfeederCopperPairsthisGrid)
P	SubfeederMaxCopperCables	Number Of Max Size Subfeeder Copper Cables		= TRUNC (IF (CumulativeCopperSubfeederPairs > MaxFeederSize, CumulativeCopperSubfeederPairs / MaxFeederSize, 0))
Q	SubfeederCopperCableSize	Subfeeder Copper Cable Size		= IF (CumulativeCopperSubfeederPairs = 0, 0, INDEX (FeederCableSize, MATCH (CumulativeCopperSubfeederPairs - (MaxFeederSize * SubfeederMaxCopperCables) , FeederCableSize, - 1) , 1))

Benchmark Cost Proxy Model Version 2.0 (BCPM2.0)
Copyright © 1997
Information contained in this document is proprietary to the BCPM2.0 Developers. The information may be used only in conjunction with the Benchmark Cost Proxy Model. Use for any other purpose is strictly prohibited.

R	SubfeederCopper UGDistance	Subfeeder Copper Underground Distance		= IF (OR (SlopeAdjSubfeederDistance = 0, SubfeederCopperCableSize = 0) , 0, IF (NumberSubfeederFibers > 0, VLOOKUP (Density, CHOOSE (FiberDepthCondition, FbrLoopHardMixTable, FbrLoopSoftMixTable, FbrLoopNormMixTable) , 2) , VLOOKUP (Density, CHOOSE (CopperDepthCondition, CopperHardMixTable, CopperSoftMixTable, CopperNormMixTable) , 2)) * SlopeAdjSubfeederDistance)
S	SubfeederUGCopperCable	Subfeeder Underground Copper Cable $		= IF (OR (SlopeAdjSubfeederDistance = 0, SubfeederCopperCableSize = 0) , 0, SubfeederCopperUGDistance * WaterCostMultiplier * CopperCostRatio * (VLOOKUP (SubfeederCopperCableSize, IF (CopperGauge = 26, UGCopper26Table, UGCopper24Table) , DensityColumnPointer, FALSE)))
T	SubfeederCopperBurDistance	Subfeeder Copper Buried Distance		= IF (OR (SlopeAdjSubfeederDistance = 0, SubfeederCopperCableSize = 0) , 0, IF (NumberSubfeederFibers > 0, VLOOKUP (Density, CHOOSE (FiberDepthCondition, FbrLoopHardMixTable, FbrLoopSoftMixTable, FbrLoopNormMixTable) , 3) , VLOOKUP (Density, CHOOSE (CopperDepthCondition, CopperHardMixTable, CopperSoftMixTable, CopperNormMixTable) , 3)) * SlopeAdjSubfeederDistance)
U	SubfeederBuriedCopperCable	Subfeeder Buried Copper Cable $		= IF (OR (SlopeAdjSubfeederDistance = 0, SubfeederCopperCableSize = 0) , 0, SubfeederCopperBurDistance * WaterCostMultiplier * CopperCostRatio * (VLOOKUP (SubfeederCopperCableSize, IF (CopperGauge = 26, BURCopper26Table, BURCopper24Table) , DensityColumnPointer, FALSE)))
V	SubfeederCopperAerialDistance	Subfeeder Copper Aerial Distance		= IF (OR (SlopeAdjSubfeederDistance = 0, SubfeederCopperCableSize = 0) , 0, IF (NumberSubfeederFibers > 0, VLOOKUP (Density, CHOOSE (FiberDepthCondition, FbrLoopHardMixTable, FbrLoopSoftMixTable, FbrLoopNormMixTable) , 4) , VLOOKUP (Density, CHOOSE (CopperDepthCondition, CopperHardMixTable, CopperSoftMixTable, CopperNormMixTable) , 4)) * SlopeAdjSubfeederDistance)
W	SubfeederAerialCopperCable	Subfeeder Aerial Copper Cable $		= IF (OR (SlopeAdjSubfeederDistance = 0, SubfeederCopperCableSize = 0) , 0, SubfeederCopperAerialDistance * WaterCostMultiplier * CopperCostRatio * (VLOOKUP (SubfeederCopperCableSize, IF (CopperGauge = 26, AERCopper26Table, AERCopper24Table) , DensityColumnPointer, FALSE)))
X	CopperSubfeederTotalMaterial	Copper Subfeeder Total Material $		= SubfeederUGCopperCable + SubfeederBuriedCopperCable + SubfeederAerialCopperCable
Y	SubfeederFiberUGDistance	Subfeeder Fiber Underground Distance		= IF (OR (SlopeAdjSubfeederDistance = 0, NumberSubfeederFibers = 0) , 0, VLOOKUP (Density, CHOOSE (FiberDepthCondition, FbrLoopHardMixTable, FbrLoopSoftMixTable, FbrLoopNormMixTable) , 2) * SlopeAdjSubfeederDistance)
Z	UGFiberSubfeederMaterial	Underground Fiber Subfeeder Material $		= IF (OR (SubfeederFiberUGDistance * SubfeederFiberSize = 0, SubfeederFiberSize = 0) , 0, SubfeederFiberUGDistance * (VLOOKUP (SubfeederFiberSize, FiberUGCostTable, DensityColumnPointer, FALSE) * FiberCostRatio))
AA	SubfeederFiberBurDistance	Subfeeder Buried Fiber Distance		= IF (OR (SlopeAdjSubfeederDistance = 0, NumberSubfeederFibers = 0) , 0, VLOOKUP (Density, CHOOSE (FiberDepthCondition, FbrLoopHardMixTable, FbrLoopSoftMixTable, FbrLoopNormMixTable) , 3) * SlopeAdjSubfeederDistance)
AB	BuriedFiberSubfeederMaterial	Buried Fiber Subfeeder Material $		= IF (SubfeederFiberBurDistance * SubfeederFiberSize = 0, 0, SubfeederFiberBurDistance * (VLOOKUP (SubfeederFiberSize, FiberBuriedCostTable, DensityColumnPointer, FALSE) * FiberCostRatio))
AC	SubfeederFiberAerialDistance	Aerial Fiber Subfeeder Distance		= IF (OR (SlopeAdjSubfeederDistance = 0, NumberSubfeederFibers = 0) , 0, VLOOKUP (Density, CHOOSE (FiberDepthCondition, FbrLoopHardMixTable, FbrLoopSoftMixTable, FbrLoopNormMixTable) , 4) * SlopeAdjSubfeederDistance)
AD	AerialFiberSubfeederMaterial	Aerial Fiber Subfeeder Material $		= IF (SubfeederFiberAerialDistance * SubfeederFiberSize = 0, 0, SubfeederFiberAerialDistance * (VLOOKUP (SubfeederFiberSize, FiberAerialCostTable, DensityColumnPointer, FALSE) * FiberCostRatio))
AE	FiberSubfeederTotalMaterial	Fiber Subfeeder Total Material $		= UGFiberSubfeederMaterial + BuriedFiberSubfeederMaterial + AerialFiberSubfeederMaterial

Benchmark Cost Proxy Model Version 2.0 (BCPM2.0)
Copyright © 1997
Information contained in this document is proprietary to the BCPM2.0 Developers. The information may be used only in conjunction with the Benchmark Cost Proxy Model. Use for any other purpose is strictly prohibited.

AF	CopperSubfeederStructureRatio	Copper Subfeeder Structure Ratio		= IF (CopperSubfeederTotalMaterial = 0, 0, IF (SubfeederMaxCopperCables > 0, Over4200, IF (AND (CopperSubfeederTotalMaterial > 0, FiberSubfeederTotalMaterial > 0) , VLOOKUP (SubfeederCopperCableSize, StructureAllocationTable, 2) , 1)))
AG	FiberSubfeederStructureRatio	Fiber Subfeeder Structure Ratio		= 1 - CopperSubfeederStructureRatio
AH	NumberSubfeederDucts	Number Subfeeder Ducts		= IF ((SubfeederUGCopperCable + UGFiberSubfeederMaterial) < 1, 0, 1 + IF (SubfeederUGCopperCable > 0, SubfeederMaxCopperCables + 1, 0) + IF (UGFiberSubfeederMaterial > 0, 1, 0))
AI	NumberSubfeederManholes	Number Subfeeder Manholes		= IF (NumberSubfeederDucts < 1, 0, CEILING (MAX (SubfeederFiberUGDistance, SubfeederCopperUGDistance) / VLOOKUP (Density, FeederSpacingTable, 2) , 1))
AJ	SubfeederStandardManholeCost	Subfeeder Standard Manhole Cost		= IF (OR (NumberSubfeederManholes = 0, NumberSubfeederDucts > 9) , 0, VLOOKUP (NumberSubfeederDucts, CHOOSE (FiberDepthCondition, ManholeHardRockTable, ManholeSoftRockTable, ManholeNormalSurfaceTable) , DensityColumnPointer))
AK	SubfeederExpandedManholeCost	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))
AL	NumberSubfeederPoles	Number Subfeeder Poles		= IF (SubfeederCopperAerialDistance < 1, 0, CEILING (MAX (SubfeederCopperAerialDistance, SubfeederFiberAerialDistance) / VLOOKUP (Density, FeederSpacingTable, 3) , 1) + 1)
AM	CopperSubfeederUGStructure	Copper Subfeeder Underground Structure $		= IF (NumberSubfeederDucts = 0, 0, CopperSubfeederStructureRatio * WaterCostMultiplier * (IF (NumberSubfeederDucts < = 9, SubfeederStandardManholeCost, SubfeederExpandedManholeCost) * NumberSubfeederManholes + MAX (SubfeederFiberUGDistance, SubfeederCopperUGDistance) * (NumberSubfeederDucts * VLOOKUP (0, ConduitDensityTable, DensityColumnPointer) + VLOOKUP (Density, CHOOSE (FiberDepthCondition, HardRockStructure, SoftRockStructure, NormalStructure) , 2))))
AN	CopperSubfeederBuriedStructure	Copper Subfeeder Buried Structure $		= IF (SubfeederCopperBurDistance + SubfeederFiberBurDistance = 0, 0, CopperSubfeederStructureRatio * WaterCostMultiplier * MAX (SubfeederFiberBurDistance, SubfeederCopperBurDistance) * IF (SubfeederFiberBurDistance = 0, VLOOKUP (Density, CHOOSE (CopperDepthCondition, HardRockStructure, SoftRockStructure, NormalStructure) , 4) , VLOOKUP (Density, CHOOSE (FiberDepthCondition, HardRockStructure, SoftRockStructure, NormalStructure) , 4)))
AO	CopperSubfeederAerialStructure	Copper Subfeeder Aerial Structure $		= IF (NumberSubfeederPoles = 0, 0, CopperSubfeederStructureRatio * WaterCostMultiplier * NumberSubfeederPoles * VLOOKUP (Density, CHOOSE (FiberDepthCondition, HardRockStructure, SoftRockStructure, NormalStructure) , 6))
AP	CopperSubfeederTotalStructure	Copper Subfeeder Total Structure $		= CopperSubfeederAerialStructure + CopperSubfeederBuriedStructure + CopperSubfeederUGStructure
AQ	FiberSubfeederUGStructure	Fiber Subfeeder Underground Structure $		= IF (NumberSubfeederDucts = 0, 0, FiberSubfeederStructureRatio * WaterCostMultiplier * (IF (NumberSubfeederDucts < = 9, SubfeederStandardManholeCost, SubfeederExpandedManholeCost) * NumberSubfeederManholes + MAX (SubfeederCopperUGDistance, SubfeederFiberUGDistance) * (NumberSubfeederDucts * VLOOKUP (0, ConduitDensityTable, DensityColumnPointer) + VLOOKUP (Density, CHOOSE (FiberDepthCondition, HardRockStructure, SoftRockStructure, NormalStructure) , 2))))
AR	BuriedFiberSubfeederStructure	Buried Fiber Subfeeder Structure $		= IF (T3 + AA3 = 0, 0, FiberSubfeederStructureRatio * WaterCostMultiplier * AA3 * IF (FiberDepthCondition = 1, VLOOKUP (Density, HardRockStructure, 4) , IF (OR (FiberDepthCondition = 2, SurfaceIndicator = 1) , VLOOKUP (Density, SoftRockStructure, 4) , VLOOKUP (Density, NormalStructure, 4))))
AS	AerialFiberSubfeederStructure	Aerial Fiber Subfeeder Structure $		= IF (NumberSubfeederPoles = 0, 0, FiberSubfeederStructureRatio * WaterCostMultiplier * NumberSubfeederPoles * VLOOKUP (Density, CHOOSE (FiberDepthCondition, HardRockStructure, SoftRockStructure, NormalStructure) , 6))

Benchmark Cost Proxy Model Version 2.0 (BCPM2.0)
Copyright © 1997
Information contained in this document is proprietary to the BCPM2.0 Developers. The information may be used only in conjunction with the Benchmark Cost Proxy Model. Use for any other purpose is strictly prohibited.

AT	FiberSubfeederTotalStructure	Fiber Subfeeder Total Structure $		= FiberSubfeederUGStructure + BuriedFiberSubfeederStructure + AerialFiberSubfeederStructure
AU	TotalCopperSubfeeder	Total Copper Subfeeder $		= CopperSubfeederTotalMaterial + CopperSubfeederTotalStructure
AV	TotalCopperSubfederCostPerPair	Total Copper Subfeeder Cost Per Pair		= IF (CumulativeCopperSubfeederPairs = 0, 0, TotalCopperSubfeeder / CumulativeCopperSubfeederPairs)
AW	CumulativeCopperSubfeederCostperPair*	Cumulative Copper Subfeeder Cost Per Pair		= IF (IsFirstSubfeederSegment, TotalCopperSubfeederCostPerPair, TotalCopperSubfeederCostPerPair + NextCumulativeCopperSubfeederCostPerPair)
AX	TotalFiberSubfeeder	Total Fiber Subfeeder $		= FiberSubfeederTotalMaterial + FiberSubfeederTotalStructure
AY	TotalFiberSubfeederCostPerPair	Total Fiber Subfeeder Cost Per Fiber		= IF (NumberSubfeederFibers = 0, 0, TotalFiberSubfeeder / NumberSubfeederFibers)
AZ	CumulativeFiberSubfeederCostperFiber*	Cumulative Fiber Subfeeder Cost Per Fiber		= IF (IsFirstSubfeederSegment, TotalFiberSubfeederCostPerPair, TotalFiberSubfeederCostPerPair + PrevCumulativeFiberSubfeederCostperFiber)

Benchmark Cost Proxy Model Version 2.0 (BCPM2.0)
Copyright © 1997
Information contained in this document is proprietary to the BCPM2.0 Developers. The information may be used only in conjunction with the Benchmark Cost Proxy Model. Use for any other purpose is strictly prohibited.

Sheet: Main&SubfeederAllocation

Col	Range Name	Column Name	Column Comment	Formula
A	CopperFdrCostthis gridbeforeallocatio n	Copper Feeder Cost This Grid Beforeallocation		= TotalMainFdrCopperCost + TotalCopperSubfeeder
B		Cumulative Main Feeder Copper Cost Per Pair		= CumulativeMainFdrCopperCostperPair
C		Cumulative Copper Subfeeder Cost Per Pair		= CumulativeCopperSubfeederCostperPair
D	TotalCopperFdrCo sttobeAllocated	Copper Feeder Cost Per pair To Be Allocated		= IF (CumulativeCopperSubfeederCostperPair > 0, CumulativeCopperSubfeederCostperPair + CumulativeMainFdrCopperCostperPair, 0)
E	CopperFdrCostAll ocatedthisGrid	Copper Feeder Cost Allocated This Grid		= SubfeederCopperPairsthisGrid * TotalCopperFdrCosttobeAllocated
F	AllocationAerialC opperCable	Allocation Of Aerial Copper Cable		= IF (CopperFdrCostthisgridbeforeallocation = 0, 0, (MainFdrSegmentAerCopperMaterial + SubfeederAerialCopperCable) * CopperFdrCostAllocatedthisGrid / CopperFdrCostthisgridbeforeallocation)
G	AllocationBuriedC opperCableinclStr ucture	Allocation Of Buried Copper Cable Incl Structure		= IF (CopperFdrCostthisgridbeforeallocation = 0, 0, (MainFdrSegmentBurCopperMaterial + MainFeederBuriedCopperStructure + SubfeederBuriedCopperCable + CopperSubfeederBuriedStructure) * CopperFdrCostAllocatedthisGrid / CopperFdrCostthisgridbeforeallocation)
H	AllocationUGCop perCable	Allocation Of UG Copper Cable		= IF (CopperFdrCostthisgridbeforeallocation = 0, 0, (MainFdrSegmentUGCopperMaterial + SubfeederUGCopperCable) * CopperFdrCostAllocatedthisGrid / CopperFdrCostthisgridbeforeallocation)
I	AllocationCopperr elatedpoles	Allocation Of Copper Related Poles		= IF (CopperFdrCostthisgridbeforeallocation = 0, 0, (MainFeederAerialCopperStructure + CopperSubfeederAerialStructure) * CopperFdrCostAllocatedthisGrid / CopperFdrCostthisgridbeforeallocation)
J	AllocationCopper RelatedConduit	Allocation Of Copper Related Conduit		= IF (CopperFdrCostthisgridbeforeallocation = 0, 0, (MainFeederUGCopperStructure + CopperSubfeederUGStructure) * CopperFdrCostAllocatedthisGrid / CopperFdrCostthisgridbeforeallocation)
K	FiberFdrCostthisgr idbeforeallocation	Fiber Feeder Cost This Grid Before Allocation	Sprint Employee: Main and Subfeeder only	= TotalFiberSubfeeder + TotalMainFdrFiberCost
L		Cumulative Main Feeder Fiber Cost Per Fiber		= CumulativeMainFdrFiberCostperFiber
M		Cumulative Fiber Subfeeder Cost Per Fiber		= CumulativeFiberSubfeederCostperFiber

Benchmark Cost Proxy Model Version 2.0 (BCPM2.0)
Copyright © 1997
Information contained in this document is proprietary to the BCPM2.0 Developers. The information may be used only in conjunction with the Benchmark Cost Proxy Model. Use for any other purpose is strictly prohibited.

N	TotalFiberFdrCost tobeallocated	Fiber Feeder Cost Per fiber To Be Allocated		= CumulativeMainFdrFiberCostperFiber + CumulativeFiberSubfeederCostperFiber
O	NextPortionCostT hisGridIf4FibersS hared*	Portion Cost This Grid If 4 Fibers Shared		= IF (TotalLinesServedInGrid = 0, 0, CHOOSE (GridPlantIndex, 0, IF (AND (NumberFibersPerGrid = 4, NumberSubfeederFibers = 4, NOT ((TotalLinesServedInGrid / FeederFill) = CumulativeDLCSLinesperSubfeeder)) , TotalFiberFdrCosttobeallocated * (TotalLinesServedInGrid / FeederFill) / CumulativeDLCSLinesperSubfeeder, TotalFiberFdrCosttobeallocated) , 0))
P	CumulativeAllocat ion4CommonFiber s*	Cumulative AllocationOf 4 Common Fibers		= IF (OR (PrevSubfeederNumber < > SubfeederNumber, PrevNumberFibersPerGrid < > PrevNumberSubfeederFibers, AND (PrevSubfeederNumber < > SubfeederNumber, PrevPortionCostThisGridIf4FibersShared = 0, PrevCumulativeAllocation4CommonFibers = 0)) , 0, O2)
Q	FiberFdrDLCScost allocatedthisgrid	Fiber Feeder DLC-S Cost Allocated This grid		= IF (PortionCostThisGridIf4FibersShared > 0, NumberFibersPerGrid * (PortionCostThisGridIf4FibersShared - IF (NumberFibersPerGrid = 0, 0, CumulativeAllocation4CommonFibers)) , NumberFibersPerGrid * PortionCostThisGridIf4FibersShared)
R	FiberFdrDLCLCos tAllocatedthisGrid	Fiber Feeder DLC-L Cost Allocated This Grid		= CHOOSE (GridPlantIndex, 0, 0, TotalFiberFdrCosttobeallocated * NumberFibersPerGrid)
S	FiberFeederallocat edthisgrid	Fiber Feeder Allocated This grid		= FiberFdrDLCSCostAllocatedThisGrid + FiberFdrDLCLCostAllocatedThisGrid
T	AllocationAerialFi berCable	Allocation Of Aerial Fiber Cable		= IF (FiberFdrCostThisGridBeforeAllocation = 0, 0, (MainFdrSegmentAerFiberMaterial + AerialFiberSubfeederMaterial) * FiberFeederAllocatedThisGrid / FiberFdrCostThisGridBeforeAllocation)
U	AllocationBuriedF iberCableinclStruc ture	Allocation Of Buried Fiber Cable Incl Structure		= IF (FiberFdrCostThisGridBeforeAllocation = 0, 0, (MainFdrSegmentBurFiberMaterial + MainFeederBuriedFiberStructure + BuriedFiberSubfeederMaterial + BuriedFiberSubfeederStructure) * FiberFeederAllocatedThisGrid / FiberFdrCostThisGridBeforeAllocation)
V	AllocationUGFibe rCable	Allocation Of UG Fiber Cable		= IF (FiberFdrCostThisGridBeforeAllocation = 0, 0, (MainFdrSegmentUGFiberMaterial + UGFiberSubfeederMaterial) * FiberFeederAllocatedThisGrid / FiberFdrCostThisGridBeforeAllocation)
W	AllocationFiberrel atedpoles	Allocation Of Fiber Related Poles		= IF (FiberFdrCostThisGridBeforeAllocation = 0, 0, (MainFeederAerialFiberStructure + AerialFiberSubfeederStructure) * FiberFeederAllocatedThisGrid / FiberFdrCostThisGridBeforeAllocation)
X	AllocationFiberRe latedConduit	Allocation Of Fiber Related Conduit		= IF (FiberFdrCostThisGridBeforeAllocation = 0, 0, (MainFeederUGFiberStructure + FiberSubfeederUGStructure) * FiberFeederAllocatedThisGrid / FiberFdrCostThisGridBeforeAllocation)
Y	TotalDLCLonFibe r	Total DLC-L on Fiber $		= FiberFdrDLCLCostAllocatedThisGrid + DLCLFiberElectronics + COTLElectronics
Z	TotalDLCSonFibe r	Total DLC-S on Fiber $		= FiberFdrDLCSCostAllocatedThisGrid + DLCSFiberElectronics + COTSElectronics
AA	SubtotalFeederCos t	Subtotal Feeder Cost		= IF (TotalLinesServedInGrid = 0, 0, CopperFdrCostAllocatedthisGrid + TotalDLCLonFiber + TotalDLCSonFiber + CopperDS1Electronics + FiberDS1Electronics)
AB	GrandTotalFeeder Cost	Grand Total Feeder Cost		= SubtotalFeederCost + TotalSubfeederPart2 + TotalDLCtoFDICableCost + FeederDistributionInterfaceCost

Benchmark Cost Proxy Model Version 2.0 (BCPM2.0)
Copyright © 1997
Information contained in this document is proprietary to the BCPM2.0 Developers. The information may be used only in conjunction with the Benchmark Cost Proxy Model. Use for any other purpose is strictly prohibited.

AC	TotalMainSubfeederPoleCost	Total Main/Subfeeder Pole Cost		= AllocationCopperRelatedPoles + AllocationFiberRelatedPoles
AD	TotalMainSubfeederConduitCost	Total Main/Subfeeder Conduit Cost		= AllocationCopperRelatedConduit + AllocationFiberRelatedConduit

Benchmark Cost Proxy Model Version 2.0 (BCPM2.0)
Copyright © 1997
Information contained in this document is proprietary to the BCPM2.0 Developers. The information may be used only in conjunction with the Benchmark Cost Proxy Model. Use for any other purpose is strictly prohibited.

Sheet: Subfeeder Part2

Col	Range Name	Column Name	Column Comment	Formula
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 (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		= 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		= 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		= 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		= 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		= 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

Benchmark Cost Proxy Model Version 2.0 (BCPM2.0)
Copyright © 1997
Information contained in this document is proprietary to the BCPM2.0 Developers. The information may be used only in conjunction with the Benchmark Cost Proxy Model. Use for any other purpose is strictly prohibited.

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		= 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	SubfeederPt2AerialStructure	Subfeeder Pt2 Aerial Structure $		= IF ((Pt2AerialCopperDistance + Pt2AerialFiberDistance) < 1, 0, ((Pt2AerialCopperDistance + Pt2AerialFiberDistance) / VLOOKUP (Density, FeederSpacingTable, 3) + 1) * 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	TotalSubfeederPart2	Total Subfeeder Part 2 $		= TotalPt2CopperMaterial + TotalPt2FiberMaterial + TotalPt2Structure

Benchmark Cost Proxy Model Version 2.0 (BCPM2.0)
Copyright © 1997
Information contained in this document is proprietary to the BCPM2.0 Developers. The information may be used only in conjunction with the Benchmark Cost Proxy Model. Use for any other purpose is strictly prohibited.

Sheet: New Distribution

Col	Range Name	Column Name	Column Comment	Formula
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	ULQBackboneand BranchCableLengt h	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, ULQBackboneandBranchCableLength = 0) , 0, CEILING (ULQQuadrantSideLength / (2 * ULQLotSideLength) , 1))
J	NumberpairsinUL quadrant	Number Of Pairs In UL Quadrant		= ULQHousingUnits * Pairsperhousingunit + IF (ULQBusinesslocations = 0, 0, IF ((ULQBusinessLines / ULQBusinesslocations) > Pairsperbusinesslocation, ULQBusinessLines, Pairsperbusinesslocation * ULQBusinesslocations))
K	PairsperULQbranc hcable	Pairs Per ULQ Branch Cable		= IF (NumberULQBranchCables = 0, 0, CEILING (NumberPairsInULQuadrant / (2 * NumberULQBranchCables) / DistributionFillFactor, 1))
L	ULQMaxSizeBran chCables	ULQ Max Size Branch Cables		= IF (PairsperULQbranchcable > MaxDistrSize, TRUNC (PairsperULQbranchcable / MaxDistrSize) , 0)
M	ULQResidualBran chCableSize	ULQ Residual Branch Cable Size		= IF (PairsperULQbranchcable = 0, 0, INDEX (DistrCableSize, MATCH (PairsperULQbranchcable - ULQMaxSizeBranchCables * MaxDistrSize, DistrCableSize, - 1) , 1))
N	ULQMaxSizeBac kboneCables	ULQ Max Size Backbone Cables		= NumberULQBranchCables * ULQMaxSizeBranchCables + IF (NumberULQBranchCables * ULQResidualBranchCableSize > MaxDistrSize, TRUNC (NumberULQBranchCables * ULQResidualBranchCableSize / MaxDistrSize) , 0)
O	ULQResidualBack boneCableSize	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))
P	ULQBranchBackb oneAerialDistance	ULQ Branch & Backbone Aerial Distance		= VLOOKUP (Density, CHOOSE (CopperDepthCondition, CopperHardMixTable, CopperSoftMixTable, CopperNormMixTable) , 4) * ULQBackboneandBranchCableLength

Benchmark Cost Proxy Model Version 2.0 (BCPM2.0)
Copyright © 1997
Information contained in this document is proprietary to the BCPM2.0 Developers. The information may be used only in conjunction with the Benchmark Cost Proxy Model. Use for any other purpose is strictly prohibited.

Q	ULQBranchAerial CableMaterial	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)))
R	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)))
S	ULQBranchBackboneBuriedDistance	ULQ Branch & Backbone Buried Distance		= VLOOKUP (Density, CHOOSE (CopperDepthCondition, CopperHardMixTable, CopperSoftMixTable, CopperNormMixTable) , 3) * ULQBackboneandBranchCableLength
T	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)))
U	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)))
V	ULQBranchBackboneUGDistance	ULQ Branch & Backbone Underground Distance		= VLOOKUP (Density, CHOOSE (CopperDepthCondition, CopperHardMixTable, CopperSoftMixTable, CopperNormMixTable) , 2) * ULQBackboneandBranchCableLength
W	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)))
X	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)))
Y	ULQBranchDucts	Number ULQ Branch Ducts		= ULQMaxSizeBranchCables + 2
Z	ULQBranchManholes	Number ULQ Branch Manholes		= NumberULQBranchCables * IF (ULQBranchBackboneUGDistance < 1, 0, ULQBranchBackboneUGDistance / VLOOKUP (Density, DistrSpacingTable, 2))
AA	ULQStandardBranchManholeCost	ULQ Standard Branch Manhole Cost		= IF (OR (ULQBranchManholes = 0, ULQBranchDucts > 9) , 0, VLOOKUP (ULQBranchDucts, CHOOSE (FiberDepthCondition, ManholeHardRockTable, ManholeSoftRockTable, ManholeNormalSurfaceTable) , DensityColumnPointer))
AB	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))

Benchmark Cost Proxy Model Version 2.0 (BCPM2.0)
Copyright © 1997
Information contained in this document is proprietary to the BCPM2.0 Developers. The information may be used only in conjunction with the Benchmark Cost Proxy Model. Use for any other purpose is strictly prohibited.

AC	ULQBackboneDucts	Number ULQ Backbone Ducts		= ULQMaxSizeBackboneCables + 2
AD	ULQBackboneManholes	Number ULQ Backbone Manholes		= IF (ULQBranchBackboneUGDistance < 1, 0, ULQBranchBackboneUGDistance / VLOOKUP (Density, DistrSpacingTable, 2))
AE	ULQStandardBackboneManholeCost	ULQ Standard Backbone Manhole Cost		= IF (OR (ULQBackboneManholes = 0, ULQBackboneDucts > 9) , 0, VLOOKUP (ULQBackboneDucts, CHOOSE (FiberDepthCondition, ManholeHardRockTable, ManholeSoftRockTable, ManholeNormalSurfaceTable) , DensityColumnPointer))
AF	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))
AG	ULQUGStructure Cost	ULQ Underground Structure Cost		= ULQBranchManholes * IF (ULQBranchDucts < = 9, ULQStandardBranchManholeCost, ULQExpandedBranchManholeCost) + ULQBackboneManholes * IF (ULQBackboneDucts < = 9, ULQStandardBackboneManholeCost, ULQExpandedBackboneManholeCost) + (ULQBranchDucts + ULQBackboneDucts) * ULQBranchBackboneUGDistance * VLOOKUP (0, ConduitDensityTable, DensityColumnPointer) + (NumberULQBranchCables + 1) * ULQBranchBackboneUGDistance * VLOOKUP (Density, CHOOSE (FiberDepthCondition, HardRockStructure, SoftRockStructure, NormalStructure) , 3)
AH	LLQHousingUnitlocations	LLQ Housing Unit Locations		= IF (HousingUnits = 0, 0, TotalHousingUnitLocations * LLQHousingUnits / HousingUnits)
AI	LLQBusinesslocations	LLQ Business Locations		= IF (TotalBusinessLines = 0, 0, CEILING (BusinessLocations * LLQBusinessLines / TotalBusinessLines, 1))
AJ	LLQLots	LLQ Lots		= LLQHousingUnitlocations + LLQBusinesslocations
AK	LLQLotSideLength	LLQ Lot Side Length		= IF (LLQLots = 0, 0, SQRT (LLQArea / LLQLots) * 5280)
AL	LLQQuadrantSideLength	LLQ Quadrant Side Length		= SQRT (LLQArea) * 5280
AM	LLQBackboneBranchCableLength	LLQ Backbone & Branch Cable Length		= IF (NumberPairsInLLQuadrant = 0, 0, IF (LLQQuadrantSideLength - (2 * LLQLotSideLength) < 0, 0, LLQQuadrantSideLength - (2 * LLQLotSideLength)))
AN	NumberLLQbranchcables	Number Of LLQ Branch Cables		= IF (OR (LLQLotSideLength = 0, LLQBackboneBranchCableLength = 0) , 0, CEILING (LLQQuadrantSideLength / (2 * LLQLotSideLength) , 1))
AO	NumberpairsinLLquadrant	Number Of Pairs In LL Quadrant		= LLQHousingUnits * Pairsperhousingunit + IF (LLQBusinesslocations = 0, 0, IF ((LLQBusinessLines / LLQBusinesslocations) > Pairsperbusinesslocation, LLQBusinessLines, Pairsperbusinesslocation * LLQBusinesslocations))
AP	PairsperLLQbranchcable	Pairs Per LLQ Branch Cable		= IF (NumberLLQBranchCables = 0, 0, CEILING (NumberPairsInLLQuadrant / (2 * NumberLLQBranchCables) / DistributionFillFactor, 1))
AQ	LLQMaxSizeBranchCables	LLQ Max Size Branch Cables		= IF (PairsperLLQbranchcable > MaxDistrSize, TRUNC (PairsperLLQbranchcable / MaxDistrSize) , 0)
AR	LLQResidualBranchCableSize	LLQ Residual Branch Cable Size		= IF (PairsperLLQbranchcable = 0, 0, INDEX (DistrCableSize, MATCH (PairsperLLQbranchcable - LLQMaxSizeBranchCables * MaxDistrSize, DistrCableSize, - 1) , 1))

Benchmark Cost Proxy Model Version 2.0 (BCPM2.0)
Copyright © 1997
Information contained in this document is proprietary to the BCPM2.0 Developers. The information may be used only in conjunction with the Benchmark Cost Proxy Model. Use for any other purpose is strictly prohibited.