



รายงานสหกิจศึกษาฉบับสมบูรณ์

เครื่องมือเตรียมข้อมูลในการกำหนดค่าสถานีฐานระบบ 3G และ 4G
Integrated data preparation tool for 3G and 4G new site
configuration

นางสาวณัฐนิชา อีระชัยกุลพานิช

ภาควิชาวิศวกรรมโทรคมนาคม
คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2560



รายงานสหกิจศึกษาฉบับสมบูรณ์

เครื่องมือเตรียมข้อมูลในการกำหนดค่าสถานีฐานระบบ 3G และ 4G
Integrated data preparation tool for 3G and 4G new site
configuration

นางสาวณัฐนิชา ชีระชัยกุลพานิช

ภาควิชาวิศวกรรมโทรคมนาคม

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2560

ชื่อโครงการสหกิจศึกษา(ภาษาไทย)

เครื่องมือเตรียมข้อมูลในการกำหนดค่าสถานีฐานระบบ 3G และ 4G

ชื่อ-สกุล นักศึกษา นางสาวณัฐนิชา ธีระชัยกุลพานิช

คณะ วิศวกรรมศาสตร์ ภาควิชา วิศวกรรมโทรคมนาคม

ชื่อ-สกุล อาจารย์นิเทศ ผศ.ดร.มนตรี คำเงิน

ชื่อ-สกุล ผู้นิเทศงาน นายธีระวุฒิ วงษ์ลา

สถานประกอบการ บริษัท แอดวานซ์ อินโฟร์ เซอร์วิส จำกัด

บทคัดย่อ

บริษัท แอดวานซ์ อินโฟร์ เซอร์วิส จำกัด มหาชน (AIS) เป็นผู้ให้บริการเครือข่ายโทรศัพท์มือถือที่ให้ความสำคัญกับคุณภาพของเครือข่ายเป็นอย่างมาก เพราะลูกค้าจะเชื่อมต่อเครือข่ายและใช้บริการอยู่ตลอดเวลา เพื่อที่จะรองรับจำนวนผู้ใช้งานให้มากขึ้น ครอบคลุมพื้นที่มากขึ้น จึงจำเป็นต้องเพิ่มสถานีฐานให้มากขึ้นด้วย สถานีฐานเป็นสถานที่ที่ติดตั้งเครื่องรับ-ส่งสัญญาณ เพื่อเชื่อมต่อสัญญาณวิทยุที่อยู่ระหว่างระบบสื่อสารแบบไร้สาย และอุปกรณ์สื่อสารไร้สาย โครงการนี้จึงจัดทำขึ้นเพื่อเพิ่มประสิทธิภาพในการเตรียมข้อมูลสำหรับการติดตั้งสถานีฐานใหม่ของกรุงเทพฯ และปริมณฑล เนื่องด้วยการเตรียมข้อมูลสำหรับสถานีฐานใหม่มีกระบวนการบางอย่างที่มีขั้นตอนการทำงานเป็นรูปแบบเดิมตลอด ดังนั้นเครื่องมือเตรียมข้อมูลในการกำหนดค่าสถานีฐานระบบ 3G และ 4G จะช่วยลดระยะเวลาในการทำงานได้ โดยการเขียนโปรแกรม VBA (Visual Basic for Application) ใน Microsoft Excel ให้ทำงานแบบอัตโนมัติ มาแก้ปัญหาในการทำงานและช่วยให้การทำงานมีความรวดเร็วมากยิ่งขึ้น

Cooperative Title : Integrated data preparation tool for 3G and 4G new site configuration

Student intern name ; Miss Natnicha Dhirachaikulpanich

Faculty : Engineering **Department** : Telecommunications Engineering

Advisor name : Asst. Prof. Dr. Montree Kumngern

Mentor name : Mr. Theerawut Wongla

Company : Advance Info Services Plc.

ABSTRACT

Advanced Info Service Plc. (AIS) is a leading provider of mobile network services that focus on network quality. Because customers will connect to the network and use the service all the time. In order to accommodate more users and cover more spaces, they need to add more Base Transceiver Station. The Base Transceiver Station is the location where the transmitter is installed for connecting the radio signal between the wireless communication system and wireless communication devices. This project is designed to improve the preparation of information for the installation of new base stations in Bangkok and its vicinity. Because of the preparation of data for new base stations, there are some processes that have worked the same way. Therefore, Integrated data preparation tool for 3G and 4G new site configuration will reduce the working time. VBA (Visual Basic for Application) programming in Microsoft Excel to run automatically for solving the problem and working faster.

กิตติกรรมประกาศ

การดำเนินโครงการในหัวข้อ เครื่องมือเตรียมข้อมูลในการกำหนดค่าสถานีฐานระบบ 3G และ 4G ของนักศึกษาสถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง สามารถสำเร็จลุล่วงไปได้ด้วยดี ได้รับความอนุเคราะห์จากอาจารย์ที่ปรึกษา ผศ.ดร.มนตรี คำเงิน ที่ได้สละเวลาในการให้คำปรึกษาและแนวทางในการดำเนินโครงการ ตลอดจนปรับปรุงแก้ไขข้อบกพร่องต่างๆ มาโดยตลอด ผู้จัดทำโครงการตระหนักถึงความทุ่มเทของอาจารย์และขอกราบขอบพระคุณเป็นอย่างสูงไว้ ณ ที่นี้ โครงการในครั้งนี้จะเกิดขึ้นไม่ได้เลย หากขาดผู้มีอุปการคุณ คุณธีระวุฒิ วงษ์ลา และพี่ๆ แผนก Radio Operation and Performance Management (RPM) ที่ได้ให้ความช่วยเหลือ ข้อคิดแนวทางการศึกษาข้อมูลต่างๆ ด้วยความใส่ใจและเต็มใจมอบให้เสมอมาจนทำให้โครงการในครั้งนี้อาจสำเร็จลุล่วงไปได้ด้วยดีตามวัตถุประสงค์ที่ตั้งไว้ สุดท้ายนี้ ขอขอบพระคุณ บริษัท แอดวานซ์ อินโฟร์ เซอร์วิส จำกัด มหาชน (AIS) ที่มอบประสบการณ์ดีๆ และให้ความอนุเคราะห์ทางผู้จัดทำตลอดการฝึกงาน รวมถึงภาควิชาวิศวกรรมโทรคมนาคมที่มอบวิชาความรู้และมอบสิ่งดีๆ ให้กับผู้จัดทำโครงการเสมอมา

นางสาว ญัฐนิชา ธีระชัยกุลพานิช

ผู้จัดทำ

สารบัญ

	หน้า
บทคัดย่อภาษาไทย	I
บทคัดย่อภาษาอังกฤษ	II
กิตติกรรมประกาศ	III
สารบัญ	IV
บทที่ 1 บทนำ	1
1.1 ความเป็นมาและความสำคัญ	1
1.2 วัตถุประสงค์	1
1.3 ขอบเขต	2
1.4 วิธีการดำเนินการ	2
1.5 ประโยชน์ที่คาดว่าจะได้รับ	2
บทที่ 2 ทฤษฎีและงานวิจัยที่เกี่ยวข้อง	3
2.1 Microsoft Excel	3
2.2 VBA (Visual Basic for Applications)	5
2.3 UMTS (Universal Mobile Telecommunication System)	12
2.4 LTE (Long-Term Evolution)	14
2.5 lub Plan	15
บทที่ 3 วิธีดำเนินการ	16
3.1 การทำงานของระบบ	16
3.2 การออกแบบระบบ	16
บทที่ 4 ผลการวิจัย	18
บทที่ 5 สรุปผลการวิจัย	30
5.1 สรุปผลการวิจัย	30
5.2 ปัญหาและอุปสรรค	31
5.3 ข้อเสนอแนะ	31
เอกสารอ้างอิง	32

สารบัญ (ต่อ)

	หน้า
ภาคผนวก	
1. Code VBA : Sheet1(process)	33
2. Code VBA : Module1	34
3. Code VBA : Module2	46
4. Code VBA : Module6	85

สารบัญตาราง

ตารางที่	หน้า
2.1 ชนิดของข้อมูล	8
2.2 Arithmetical Operators	11
2.3 String operator	11
2.4 Comparison operation	11
2.5 Logical operator	12

สารบัญญภาพ

ภาพที่	หน้า
2.1 Microsoft Excel	3
2.2 คุณสมบัติของ Microsoft Excel	4
2.3 ประโยชน์ของ Microsoft Excel	5
2.4 Microsoft Visual Basic for Application	5
2.5 แถบ Developer	6
2.6 Insert module	6
2.7 Module	6
2.8 องค์ประกอบของหน้าต่าง VBE	7
2.9 ตัวแปรแบบ Local	9
2.10 ตัวแปรแบบ Public	10
2.11 ตัวแปรอาร์เรย์	10
2.12 โครงสร้างของ UMTS	14
2.13 การรวมฟังก์ชันเครือข่ายหลักหลายแบบเพื่อประสิทธิภาพที่เหนือกว่า	14
3.1 บล็อกไดอะแกรมของเครื่องมือเตรียมข้อมูลในการกำหนดค่าสถานีฐานระบบ 3G และ 4G	16
3.2 Flow chart ของเครื่องมือเตรียมข้อมูลในการกำหนดค่าสถานีฐานระบบ 3G และ 4G	17
4.1 ข้อมูลภายในไฟล์ lub Plan	18
4.2 หน้าต่าง GUI	19
4.3 หน้า GUI เมื่อเบราวซ์ไฟล์และใส่ชื่อสถานีฐาน	19

สารบัญญภาพ (ต่อ)

ภาพที่	หน้า
4.4 ข้อมูลในไฟล์ UMTS (3G) cell parameter	20
4.5 ข้อมูลในไฟล์ LTE (4G) cell parameter	20
4.6 หน้าต่างแสดงข้อความว่า “Comfirm to generate?”	21
4.7 หน้าต่างแสดงประเภทของ BSC	21
4.8 Worksheet 1 : Base Station Transport Data	21
4.9 Worksheet 2 : UMTS Cell	22
4.10 Worksheet 3 : LTE Cell	22
4.11 Worksheet 4 : Multicarrier Cell Group	23
4.12 Worksheet 5 : InterRatHoUtranGroup	23
4.13 Worksheet 6 : InterRatHoCommGroup	24
4.14 Worksheet 7 : UtranNFreq	24
4.15 Worksheet 8 : TcelpMapping	25
4.16 หน้าต่างแสดงข้อความเตือน “Error!! Please check your data.”	25
4.17 Error Report	26
4.18 ข้อมูลที่ผิดพลาดในไฟล์ UMTS cell parameter	27
4.19 ข้อมูลที่ผิดพลาดในไฟล์ LTE cell parameter	28
4.20 ซีทแรกของ LTE cell parameter ที่ชื่อว่า LTE2100	28
4.21 หน้าต่างแสดงข้อความเตือน “The first sheet of ODD isn’t LTE1800!!”	29

สารบัญญภาพ (ต่อ)

ภาพที่	หน้า
4.22 ข้อมูลในชีท Data	29

บทที่ 1

บทนำ

1.1 ความเป็นมาและความสำคัญ

โทรศัพท์มือถือในยุคแรกนั้น เริ่มจากการใช้สถานีฐานแห่งเดียวที่ตั้งอยู่ในใจกลางเมือง สถานีนี้มีอุปกรณ์รับส่งวิทยุที่มีกำลังส่งสูง เพื่อให้สามารถครอบคลุมพื้นที่ให้กว้างที่สุดเท่าที่เป็นไปได้ เพราะอุปกรณ์รับส่งมีราคาแพงและการใช้สถานีฐานจำนวนมากจะทำให้สิ้นเปลืองค่าใช้จ่ายในการวางระบบ เมื่อผู้ใช้บริการเริ่มมีจำนวนมากขึ้น ระบบโทรศัพท์เคลื่อนที่ในยุคนี้ไม่สามารถรองรับได้เพียงพอ จึงเปลี่ยนมาใช้โทรศัพท์เซลลูลาร์ซึ่งเป็นระบบโทรศัพท์เคลื่อนที่ระบบใหม่ พัฒนาขึ้นมาให้เหมาะสมกับการใช้งานในระบบวิทยุโทรศัพท์เคลื่อนที่ในปัจจุบัน สามารถครอบคลุมพื้นที่ให้บริการได้กว้างมากขึ้น ขยายขอบเขตการให้บริการได้อย่างต่อเนื่อง

แผนก Radio Operation and Performance Management (RPM) ในฝ่าย Radio and Wifi Operation (RWO) มีหน้าที่ในการดูแลสถานีฐานของกรุงเทพฯ และปริมณฑล การติดตั้งสถานีฐานใหม่เป็นส่วนหนึ่งของการทำงานในแผนก การเตรียมข้อมูลในการขึ้นสถานีฐานใหม่จำเป็นต้องใช้ข้อมูลจากหลายๆ ฝ่าย ซึ่งมีข้อมูลจำนวนมากและมีความซับซ้อน เนื่องด้วยการเตรียมข้อมูลสำหรับสถานีฐานใหม่มีกระบวนการบางอย่างที่มีขั้นตอนการทำงานเป็นรูปแบบเดิมตลอด ดังนั้นถ้ามีเครื่องมือในการเตรียมข้อมูลสำหรับสถานีฐานใหม่ของ 3G และ 4G ที่ช่วยลดระยะเวลาในการทำงานได้ ซึ่งสามารถใช้ประโยชน์จากการเขียนโปรแกรม VBA (Visual Basic for Application) ใน Microsoft Excel ให้ทำงานแบบอัตโนมัติ มาแก้ปัญหาในการทำงาน และช่วยให้การทำงานมีประสิทธิภาพมากยิ่งขึ้น จึงเกิดเป็นโครงการนี้ขึ้น

1.2 วัตถุประสงค์

1. เพื่อลดกระบวนการของการเตรียมข้อมูลในการขึ้นสถานีฐานใหม่ให้มีความรวดเร็วยิ่งขึ้น
2. เพื่อลดระยะเวลาที่ใช้ในการจัดเตรียมข้อมูลสำหรับการกำหนดค่าสถานีฐานใหม่
3. เพื่อลดความผิดพลาดที่อาจเกิดขึ้นจากการกระบวนการเตรียมข้อมูลแบบเดิม
4. เพื่อเพิ่มประสิทธิภาพของการทำงานและประสิทธิผลของผลลัพธ์ในการทำงานให้มากยิ่งขึ้น

1.3 ขอบเขต

1. โครงการนี้จัดทำขึ้นเพื่อจัดการข้อมูลที่ได้รับจากหลายๆแผนก สำหรับการขึ้นสถานีฐานใหม่
2. การใช้งานโปรแกรม VBA ในการจัดการกระบวนการต่างๆให้ได้มาซึ่งไฟล์ excel สำหรับการขึ้นสถานีฐานใหม่ โดยสามารถใช้งานได้ หนึ่งชื่อสถานีฐานต่อหนึ่งไฟล์ cell parameter ของ 3G และหนึ่งไฟล์ cell parameter ของ 4G เท่านั้น
3. หากมีสองชื่อสถานีฐานอยู่ในไฟล์ cell parameter เดียวกันก็สามารถใช้งานได้ แต่การใส่ชื่อสถานีฐานจะทำได้เพียงครั้งละหนึ่งชื่อ

1.4 วิธีการดำเนินการ

1. เรียนรู้กระบวนการแบบเดิมของการเตรียมไฟล์ excel สำหรับการขึ้นสถานีฐานใหม่
2. ทดลองทำกระบวนการแบบเดิม เพื่อให้รู้ถึงขั้นตอนและวิธีการทำในการเตรียมข้อมูล
3. ศึกษาการเขียนโปรแกรม VBA (Visual Basic for Application) ใน Microsoft Excel

1.5 ประโยชน์ที่คาดว่าจะได้รับ

1. ผู้ใช้งานสามารถเตรียมข้อมูลสำหรับสถานีฐานใหม่ได้ แม้จะไม่เคยเรียนรู้กระบวนการแบบเดิม
2. ผู้จัดทำโครงการได้เรียนรู้กระบวนการและขั้นตอนการทำงานแบบเดิมที่ใช้ในการเตรียมข้อมูล
3. ผู้จัดทำโครงการสามารถเขียนโปรแกรม VBA ในระดับเบื้องต้นได้
4. ผู้ใช้งานสามารถเตรียมข้อมูลได้ โดยใช้ระยะเวลาที่น้อยลงกว่าเดิม

บทที่ 2

แนวคิดและทฤษฎีที่เกี่ยวข้อง

2.1 Microsoft Excel

เป็นโปรแกรมประเภท สเปรดชีต (Spreadsheets) หรือโปรแกรมตารางงาน ซึ่งจะเก็บข้อมูลต่าง ๆ ลงบนแผ่นตารางงาน คล้ายกับการเขียนข้อมูลลงไปในสมุดที่มีการตีช่องตารางทั้งแนวนอนและแนวตั้ง ซึ่งช่องตารางแต่ละช่องจะมีชื่อประจำแต่ละช่อง ทำให้ง่ายต่อการป้อนข้อมูล การแก้ไขข้อมูล สะดวกต่อการคำนวณและการนำข้อมูลไปประยุกต์ใช้ สามารถจัดข้อมูลต่าง ๆ ได้อย่างเป็นหมวดหมู่และเป็นระเบียบมากยิ่งขึ้น อีกทั้งยังเป็นโปรแกรมที่มีความสามารถรอบด้าน ไม่ว่าจะเป็นด้านงานวิเคราะห์ หรือการคำนวณ รวมถึงนำข้อมูลในตารางมาแสดงผลในรูปแบบที่สามารถเข้าใจข้อมูลนั้นลึกซึ้งมากยิ่งขึ้น เช่น สร้างกราฟ หรือจะตารางที่ทำให้เปลี่ยนมุมมองไปมาได้อย่างง่ายดาย



ภาพที่ 2.1 Microsoft Excel

2.1.1 คุณสมบัติของ Microsoft Excel

1. ความสามารถด้านการคำนวณ Excel สามารถป้อนสูตรการคำนวณทางคณิตศาสตร์ เช่น บวก ลบ คูณ หาร เป็นต้น รวมทั้งสูตรคำนวณด้านอื่น ๆ และจุดเด่นของการคำนวณคือผลลัพธ์ของการคำนวณจะเปลี่ยนแปลงตาม เมื่ออินพุตที่นำมาเปลี่ยนค่า ทำให้ไม่ต้องเสียเวลาเปลี่ยนแปลงค่าผลการคำนวณใหม่
2. ความสามารถด้านการใช้ฟังก์ชัน นอกจากการป้อนสูตรคูณทางคณิตศาสตร์แล้ว Excel ยังสามารถป้อนฟังก์ชันอื่น ๆ ได้อีก เช่น ฟังก์ชันเกี่ยวกับตัวอักษร ตัวเลข วันที่ ฟังก์ชันเกี่ยวกับการเงิน หรือการตัดสินใจ

3. ความสามารถในการสร้างกราฟ Excel สามารถนำข้อมูลที่ป้อนลงในตารางมาสร้างเป็นกราฟได้ทันที มีรูปกราฟให้เลือกใช้งานหลายรูปแบบตามความเหมาะสม เช่น กราฟแท่ง แสดงยอดขายแต่ละเดือน กราฟวงกลม แสดงส่วนแบ่งการตลาด เป็นต้น

4. ความสามารถในการตกแต่งตารางข้อมูล Excel สามารถตกแต่งตารางข้อมูลหรือกราฟข้อมูลด้วยภาพสีและรูปแบบตัวอักษรต่าง ๆ เพื่อให้เกิดความสวยงามและแยกแยะข้อมูลได้ง่ายขึ้น

5. ความสามารถในการจัดเรียงลำดับ Excel สามารถคัดเลือกเฉพาะข้อมูลที่ต้องการมาวิเคราะห์ได้

6. ความสามารถในการพิมพ์งานออกทางเครื่องพิมพ์ Excel สามารถพิมพ์งานทั้งข้อมูลและรูปภาพหรือกราฟออกทางเครื่องพิมพ์ได้ทันที ซึ่งทำให้ง่ายต่อการสร้างรายงาน

7. ความสามารถในการแปลงข้อมูลในตารางให้เป็นเว็บเพจเพื่อนำมาแสดงในโฮมเพจ



ภาพที่ 2.2 คุณสมบัติของ Microsoft Excel

2.1.2 ประโยชน์ของ Microsoft Excel

1. ความสามารถด้านการคำนวณ โปรแกรม Microsoft Excel สามารถป้อนสูตรการคำนวณทางคณิตศาสตร์ เช่น บวก ลบ คูณ ทหาร เป็นต้น

2. สร้างรายงานสรุปผลในมุมมองต่างๆ เช่น ตารางสรุปยอดขาย ตารางสรุปข้อมูลสินค้า สรุปงบดุล สรุปแผนการผลิต สรุปข้อมูล ขาดลามาสาย ของพนักงาน เป็นต้น

3. มีความสามารถในการจัดการข้อมูลและฐานข้อมูล ซึ่งเป็นกลุ่มของข้อมูลข่าวสาร ที่ถูกรวบรวมเข้าไว้ด้วยกันในตารางที่อยู่ใน Worksheet ลักษณะของการเก็บข้อมูลเพื่อใช้เป็นฐานข้อมูลบนโปรแกรมตารางงานจะเก็บข้อมูลในรูปแบบของตาราง โดยแต่ละแถวของรายการจะเป็นระเบียบหรือเรคอร์ด (Record) และคอลัมน์จะเป็นฟิลด์ (Field)

4. ใช้การจัดรูปแบบตามเงื่อนไขกับข้อมูลของคุณได้สะดวกมากยิ่งขึ้นเพื่อค้นหารูปแบบ และเน้นแนวโน้มในข้อมูลของคุณ มีโครงสร้างแบบใหม่ซึ่งประกอบไปด้วยแถบไลส์ สีตแมป แถบข้อมูล และไอคอนตัวบ่งชี้ประสิทธิภาพ

5. สร้างกราฟ นำเสนอข้อมูล ในรูปแบบต่างๆ ไม่ว่าจะเป็นกราฟแท่ง กราฟเส้น วงกลม จุด ทั้ง 2 มิติ และ 3 มิติ มีรูปแบบต่างๆ มากมาย

6. สร้างตารางทำงาน จัดตารางสวยงาม ในรูปแบบต่างๆ ได้หลายรูปแบบ มีความถูกต้อง แม่นยำสูงมาก



ภาพที่ 2.3 ประโยชน์ของ Microsoft Excel

2.2 VBA (Visual Basic for Applications)

เป็นโปรแกรมที่มีลักษณะคล้ายคลึงกับภาษาของโปรแกรมอื่นๆ รวมถึงการพิมพ์ค่าต่างๆ การวนลูป การประกาศตัวแปร และอาร์เรย์ VBA กลายมาจาก VB (Visual Basic) แต่เรามุ่งไปที่แอปพลิเคชัน ในโปรแกรมที่ใช้เท่านั้น ภาษานี้อนุญาตให้จัดการในโปรแกรมของ Microsoft เช่น Excel, Word, Powerpoint, Access เป็นต้น

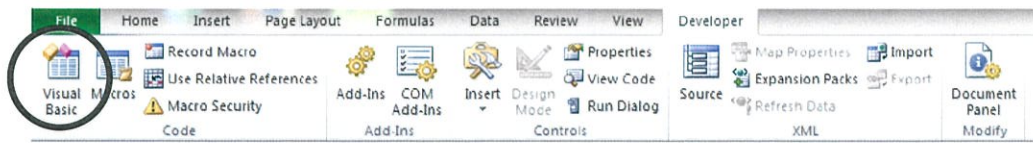


ภาพที่ 2.4 Microsoft Visual Basic for Application

2.2.1 วิธีการเปิดหน้าต่าง VBE (Visual Basic Editor)

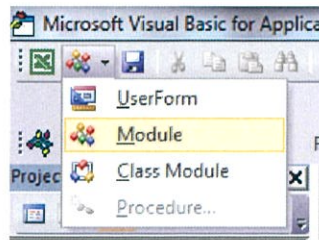
1. คลิกที่ File > Option
2. เลือก Tab Customize Ribbon > ทำเครื่องหมายที่ Developer > OK

3. แถบ Developer จะเห็นว่ามือเครื่องมือสำหรับการบันทึก Macro และการเขียน VBA รวมทั้งการสร้าง Control ต่างๆ บน Worksheet เพื่อสะดวกในการใช้งาน



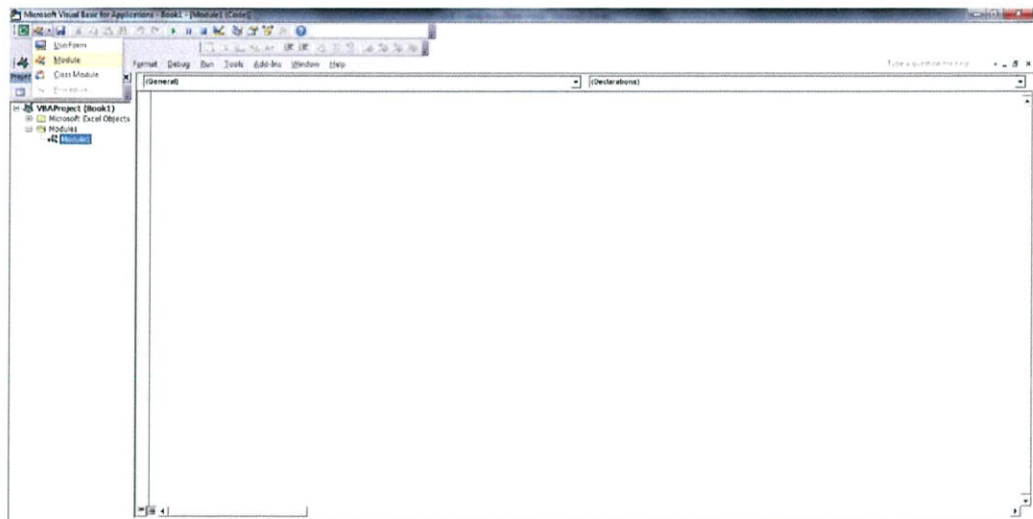
ภาพที่ 2.5 แถบ Developer

4. คลิกที่ Visual Basic > insert module > Module



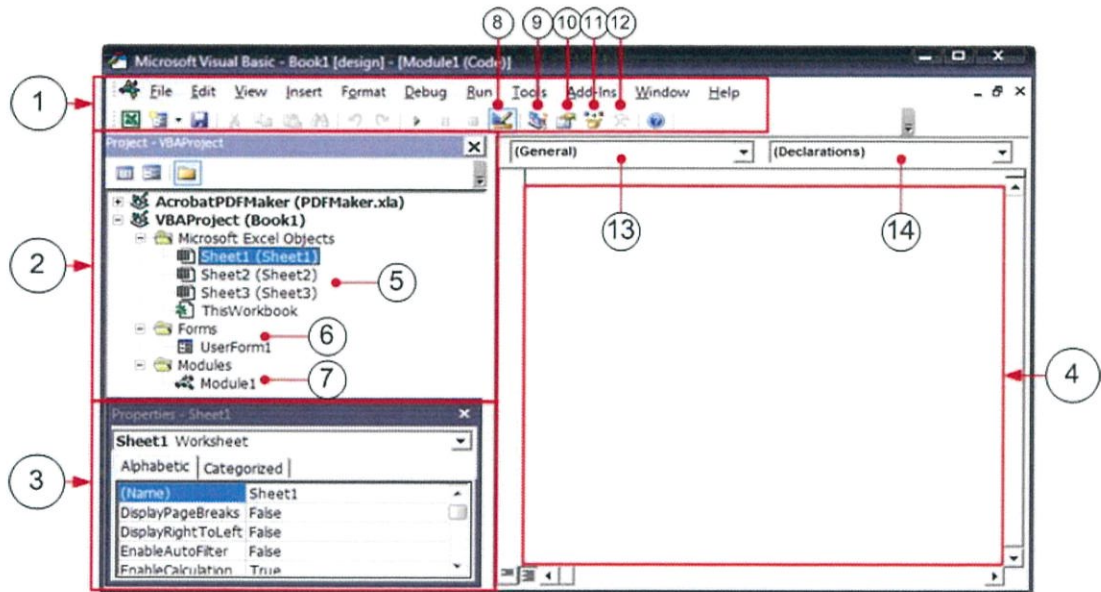
ภาพที่ 2.6 Insert module

5. ได้ Module สำหรับการเขียนโค้ด VBA



ภาพที่ 2.7 Module

2.2.2 องค์ประกอบของหน้าต่าง VBE



ภาพที่ 2.8 องค์ประกอบของหน้าต่าง VBE

1. เมนูบาร์
2. Project Explorer คือ หน้าต่างของโปรเจค
3. Properties Window คือ หน้าต่างคุณสมบัติของ object ต่างๆ
4. พื้นที่สำหรับเขียน พิมพ์และแก้ไขโค้ด
5. Object ของสมุดงาน (workbook)
6. ฟอรัมต่างๆ ที่สร้างขึ้น
7. Module คือ ที่เก็บรหัสคำสั่งต่างๆที่เขียนขึ้นมา
8. ปุ่ม Design mode
9. ปุ่มใช้แสดงหน้าต่าง Project Explorer
10. ปุ่มใช้แสดงหน้าต่าง Properties Window
11. ปุ่มใช้แสดงหน้าต่าง Object Browser
12. ปุ่ม ToolBox คือ กล่องเครื่องมือต่างๆ ที่ใช้ในการสร้างองค์ประกอบของฟอร์ม
13. ปุ่ม DropDownList ของ Object
14. ปุ่ม DropDownList ของ Procedure

2.2.3 วิธีการเขียนโปรแกรมเบื้องต้น

2.2.3.1 ชนิดของข้อมูล

Visual Basic มีชนิดของข้อมูลหลายชนิด ไม่ว่าจะเป็นตัวเลขจำนวนเต็ม ตัวเลขที่มีทศนิยม ข้อความ ตัวเลขทางการเงิน ค่าทางตรรกะ เป็นต้น ข้อมูลแต่ละชนิด จะใช้พื้นที่ในการเก็บไม่เท่ากัน รวมถึงความเร็วในการประมวลผลก็แตกต่างกันด้วย สามารถแบ่งชนิดของข้อมูลที่ใช้กัน Visual Basic ได้ดังตารางต่อไปนี้

ตารางที่ 2.1 ชนิดของข้อมูล

ชนิดข้อมูล	รายละเอียด	หน่วยความจำ
Boolean	เก็บค่าทางตรรกะที่ได้มี 2 ค่า คือ True, False	2 Bytes
Byte	เก็บค่าเลขจำนวนเต็มตั้งแต่ 0-255	1 Byte
Date	ใช้สำหรับเก็บวันที่และเวลา	8 Bytes
Integer	เก็บค่าเลขจำนวนเต็มที่มีค่าระหว่าง - 32768 ถึง 32767	2 Bytes
Long	ใช้เก็บเลขจำนวนเต็มที่มีค่าระหว่าง -2,147,483,648 ถึง 2,147,483,647	4 Bytes
String	ใช้เก็บตัวอักษร, ตัวเลข, ช่องว่าง และ เครื่องหมายวรรคตอน	1 Bytes
Variant	ข้อมูลพิเศษสามารถเก็บข้อมูลได้ทุกชนิด	16 Bytes

2.2.3.2 การประกาศค่าตัวแปร (Variable Declaration)

ก่อนที่จะใช้งานตัวแปร หรือค่าคงที่ทุกครั้ง ควรประกาศตัวแปร (variable declaration) ก่อน เพื่อให้ Visual Basic รู้ว่า ตัวแปรที่ต้องการใช้งาน ใช้แทนข้อมูลชนิดใดถึงแม้ว่า Visual Basic อนุญาตให้ใช้งานตัวแปรได้ โดยไม่ต้องประกาศตัวแปร แต่ตัวแปรที่ได้จะใช้ทรัพยากรระบบ มากเกินความจำเป็น รวมถึงประมวลผลได้ช้า เพราะจะเป็นตัวแปรที่สามารถแทนข้อมูลได้ทุกชนิด ซึ่ง Visual Basic เรียกว่า ตัวแปรชนิด Variant ซึ่งควรหลีกเลี่ยงการใช้งานตัวแปรชนิดนี้

รูปแบบคำสั่งการประกาศตัวแปรของ Visual Basic

Dim varname As datatypes

Dim คือ คำสั่ง (statements) สำหรับประกาศตัวแปร

varname คือ ชื่อของตัวแปรที่ต้องการประกาศ

As คือ ส่วนที่บอกให้ Visual Basic ทราบว่าต้องการกำหนดชนิดของข้อมูล

datatypes คือ ชนิดของข้อมูลที่ Visual Basic สนับสนุน

กรณีที่ต้องการกำหนดให้มีการประกาศตัวแปรทุกครั้งก่อนที่จะมีการเรียกใช้ตัวแปร เพื่อหลีกเลี่ยงการใช้ตัวแปรชนิด Variant ให้พิมพ์คำสั่ง Option Explicit ไว้ข้างบนสุดก่อนพิมพ์คำสั่งอื่น ๆ

2.2.3.3 ขอบเขตของตัวแปร (Scope of Variable)

- ตัวแปรแบบ Local

หมายถึง ตัวแปรที่ประกาศขึ้นมาให้สามารถเรียกใช้งานได้ ในเฉพาะโปรแกรมเมอร์ที่ประกาศเท่านั้น

ตัวอย่าง

```
Private Sub AddNum_Click()  
Dim x As Integer  
Dim y As Integer  
x = 5  
y = 6  
x = x + y  
End Sub  
.....  
Private Sub DelNum_Click()  
Dim x As Integer  
Dim y As Integer  
x = 3  
y = 2  
End Sub
```

ภาพที่ 2.9 ตัวแปรแบบ Local

จากตัวอย่าง ตัวแปร x และ y ถูกประกาศใน Sub AddNum และ Sub DelNum ซึ่ง x และ y ใน AddNum จะเป็นคนละตัวกับ x และ y ใน DelNum

- ตัวแปรแบบ Public

หมายถึง ตัวแปรที่ประกาศขึ้นในส่วนบนหลัง Option Explicit ทำให้สามารถเรียกใช้งานได้ทุกโปรแกรมเมอร์ในฟอร์มนั้น กรณีที่ประกาศตัวแปรแบบ Public ใน Module จะทำให้ตัวแปรนั้นสามารถเรียกใช้งานได้จากทุกฟอร์ม

ตัวอย่าง

```
Option Explicit
Dim x As Integer
Dim y As Integer
Private Sub AddNum_Click()
x = 5
y = 6
x = x + y
End Sub

Private Sub DelNum_Click()
x = x - y
End Sub
```

ภาพที่ 2.10 ตัวแปรแบบ Public

จากตัวอย่าง ตัวแปร x และ y ถูกประกาศในโค้ดจาก Option Explicit ก่อน Sub AddNum ทำให้ตัวแปร x และ y ใน Sub AddNum และ Sub DelNum เป็นตัวแปรตัวเดียวกัน

2.2.3.4 ตัวแปรอาร์เรย์ (Array)

เป็นกลุ่มของตัวแปรที่ประกาศขึ้นมา โดยใช้ชื่อของตัวแปรเดียวใช้ค่า index ในการอ้างถึง ประโยชน์ของตัวแปรชนิดนี้คือ กรณีที่ต้องการมีการใช้ตัวแปรจำนวนมาก การประกาศตัวแปรสามารถประกาศเพียงชื่อเดียว ลดความซ้ำซ้อนของตัวแปร และทำให้ง่ายต่อการเรียกใช้

ก่อนการใช้งานอาร์เรย์ จำเป็นต้องประกาศตัวแปรอาร์เรย์มาก่อน โดยการใช้คำสั่ง Dim เช่นเดียวกับกับการประกาศตัวแปรปกติ แต่สำหรับอาร์เรย์จะต้องมีการกำหนดขนาดด้วย โดยการใช้เครื่องหมาย () เพื่อบ่งบอกว่ามันเป็นตัวแปรอาร์เรย์

ตัวอย่าง

```
Dim number(4) As Integer

number(0) = 10
number(1) = 20
number(2) = 30
number(3) = 40
number(4) = 50
```

ภาพที่ 2.11 ตัวแปรอาร์เรย์

คำสั่ง dim number(4) as integer นี้เป็นการประกาศตัวแปรอาร์เรย์ที่ชื่อ number กำหนดขนาดโดย (4) หมายความว่าขนาดที่แท้จริงจะบวกไปอีก 1 เพราะค่า index ของอาร์เรย์ในภาษา Visual Basic นั้นจะเริ่มต้นที่ 0 ถึง 4 ดังนั้นมันจึงมีขนาดเป็น 5 นั่นเอง ต่อมาเป็นการกำหนดค่าให้กับอาร์เรย์แต่ละตำแหน่งผ่านทาง index ของมัน โดย

ใส่ตัวเลข index ภายในเครื่องหมาย () เหมือนที่เห็น number(0) เป็นข้อมูลในตำแหน่งแรกของอาร์เรย์ และ number(4) เป็นข้อมูลตำแหน่งสุดท้ายของอาร์เรย์

2.2.3.5 ตัวดำเนินการ (Operation)

- Arithmetical Operators หมายถึง เครื่องหมายคำนวณทางคณิตศาสตร์

ตารางที่ 2.2 Arithmetical Operators

Operator	Description	Example	Result
s			
+	บวก	2+2	4
-	ลบ	5-3	2
*	คูณ	2*3	6
/	หาร	20/10	2
\	หารเอาผลลัพธ์เป็นจำนวนเต็ม (ปิดเศษทิ้ง)	25/6	4
Mod	หารและส่งคืนค่าเศษแบบปิดเป็นจำนวนเต็ม	20 Mod 3	2
^	ยกกำลัง	4^2	16

- ตัวดำเนินการเชิงข้อความ(String operator) เป็นการเชื่อมโยงข้อความเข้าด้วยกัน

ตารางที่ 2.3 String operator

Operators	Description	Example	Result
&	ใช้รวมตัวอักษร	"สวัสดี"&"ครับ"	"สวัสดีครับ"
+	ใช้รวมตัวอักษร	"สวัสดี"+"ครับ"	"สวัสดีครับ"

- Comparison operation ใช้เพื่อการเปรียบเทียบข้อมูล

ตารางที่ 2.4 Comparison operation

Operators	Description	Example	Result
>	มากกว่า	10>8	จริง (True)
<	น้อยกว่า	10<8	เท็จ (False)
>=	มากกว่าหรือเท่ากับ	20>=10	จริง (True)

<=	น้อยกว่าหรือเท่ากับ	10<=20	จริง (True)
<>	ไม่เท่ากับ	5<>4	จริง (True)
=	เท่ากับ	5=7	เท็จ (False)

- Logical operator ตัวดำเนินการเชิงตรรก (Logical operator) เป็นเครื่องหมายที่ใช้เปรียบเทียบข้อมูล โดยค่าที่ได้จะเป็นค่าบูลีนคือ ค่าจริง(True) และเท็จ (False)

ตารางที่ 2.5 Logical operator

Operators	Description
AND	หมายถึง “และ” จะเป็นจริงเมื่อค่าที่ใช้เปรียบเทียบทั้ง 2 ค่า เป็นจริงทั้งคู่
OR	หมายถึง “หรือ” จะเป็นจริงเมื่อค่าที่ใช้เปรียบเทียบทั้ง 2 ค่า เป็นจริงทั้งคู่ หรือเป็นจริงเพียงค่าใดค่าหนึ่ง
NOT	หมายถึง “ปฏิเสธ” เป็นการแปลงค่าตรงกันข้าม จากจริงจะเป็นเท็จ และจากเท็จจะเป็นจริง

2.2.4 ประโยชน์ของ VBA

1. ลดระยะเวลาในการทำงานซ้ำๆ ลง
2. เพิ่มประสิทธิภาพในการทำงาน
3. ช่วยให้ผู้ที่ไม่มีความรู้เกี่ยวกับ Excel มาก ทำกระบวนการที่ซับซ้อนได้ โดยการรันมาโครที่บันทึกไว้ได้
4. ช่วยให้ข้อมูลที่ทำงานเป็นรูปแบบเดียวกัน เพราะจะได้ผลลัพธ์เหมือนกัน แม้ต่างคนจะต่างใช้งาน

2.3 UMTS (Universal Mobile Telecommunication System)

ระบบการสื่อสารโทรคมนาคมเคลื่อนที่ เป็นมาตรฐานที่ออกแบบมาสำหรับผู้ให้บริการเครือข่ายโทรศัพท์เคลื่อนที่ได้นำไปพัฒนาจากยุค 2G ไปสู่ ยุค 3G อย่างเต็มตัว คือ มีพัฒนาการมาจากเครือข่าย GSM, GPRS และ EDGE มีเทคโนโลยีหลักที่มีการยอมรับใช้งานทั่วโลกคือมาตรฐาน Wideband Code Division Multiple Access (W-CDMA) โดยมีจุดมุ่งหมายเพื่อตอบสนองความต้องการใช้งานด้านการรับ-ส่งข้อมูลที่มากขึ้นของลูกค้า การเข้าถึงเครือข่ายแบบไร้สาย สามารถทำได้ด้วยอุปกรณ์หลากหลาย เช่น จาก

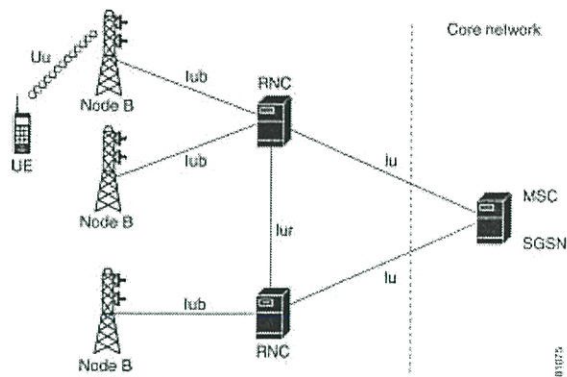
คอมพิวเตอร์ จากเครื่องใช้ไฟฟ้าอื่น ระบบยังคงให้การเข้าช่องสัญญาณเป็นแบบ CDMA ซึ่งสามารถบรรจุช่องสัญญาณเสียงได้มากกว่า แต่ใช้แบบแถบกว้าง (wideband) ในระบบนี้จึงเรียกอีกอย่างหนึ่งว่า W-CDMA มีความสามารถในการนำเอนข้อมูล ใช้งานด้านมัลติมีเดีย ส่งผ่านข้อมูลทั้งภาพและเสียงในระบบไร้สายด้วยความเร็วที่สูง เครือข่าย UMTS จะมีความเร็วในการรับส่งข้อมูลสูงถึง 2 Mbit/sec

มาตรฐานของ UMTS ในปัจจุบันนั้นมีการเผยแพร่ออกมาแล้ว 4 มาตรฐานด้วยกัน โดยหน่วยงาน 3GPP (3G Partnership Project) รับผิดชอบในการออกแบบมาตรฐานต่าง ๆ ซึ่งประกอบไปด้วย

- Release 99 เป็นมาตรฐานใช้งานที่เพิ่มเติมจากเครือข่าย GPRS และ EDGE โดยจะมีการเพิ่มเติมอุปกรณ์ในส่วนของ BSS (Base Station Subsystem) ซึ่งเป็นส่วนที่ดูแลการติดต่อสื่อสารระหว่างเครื่องโทรศัพท์เคลื่อนที่ของ ผู้ใช้บริการกับเครือข่ายของผู้ให้บริการ โดยกลุ่มของอุปกรณ์ที่เพิ่มเติมขึ้นมานี้มีชื่อเรียกว่า UTRAN (UMTS Terrestrial Radio Access Network)
- Release 4 เป็นมาตรฐานที่เพิ่มเติมในส่วนของ Core-Network โดยจะมีการนำเครือข่ายแบบ ATM (Asynchronous Transfer Mode) และ IP ซึ่งเป็นการรับ-ส่งข้อมูลแบบเป็น Packet เข้ามาใช้งานแทนเครือข่ายแบบ Circuit Switched ที่ใช้งานอยู่ในเครือข่าย GSM ในปัจจุบัน
- Release 5 เป็นมาตรฐานที่เพิ่มเติมในส่วนของ IMS (IP Multimedia Service) โดยการทำงานของ IMS จะช่วยให้การใช้งานแบบ Multimedia ในลักษณะของ Person to Person มีประสิทธิภาพที่สูงขึ้น
- Release 6 เป็นมาตรฐานที่ไม่ได้มีการเปลี่ยนแปลงระบบมากนัก เพียงแต่เพิ่มความสามารถในการทำงานของการจดจำคำพูด (Speech Recognition), Wi-Fi / UMTS inter-working (การสื่อสารระหว่างเครือข่าย Wireless LAN กับเครือข่ายโทรศัพท์เคลื่อนที่)

2.3.1 โครงสร้างหลักของมาตรฐาน UMTS

- ♣ User Equipment : โทรศัพท์เคลื่อนที่ หรือสถานีเคลื่อนที่
- ♣ NodeB : สถานีฐานในเน็ตเวิร์ค UMTS ซึ่งสื่อสารโดยตรงกับโทรศัพท์เคลื่อนที่ปลายทาง ด้วยคลื่นความถี่วิทยุตัวรับและตัวส่ง
- ♣ RNC (Radio Network Controller) : ควบคุมการใช้ทรัพยากรของสถานีฐานดูแลการต่อเชื่อมกับเครื่องลูกข่าย

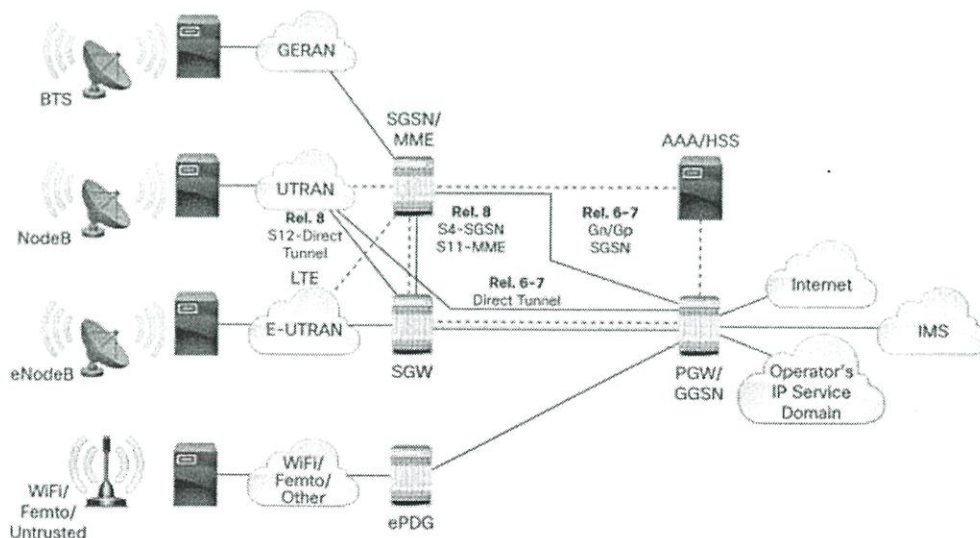


ภาพที่ 2.12 โครงสร้างของ UMTS

2.4 LTE (Long-Term Evolution)

เป็นเทคโนโลยีหนึ่งที่ถูกนำมาทดลองใช้ในยุค 4G โดยเกิดจากความร่วมมือของ 3GPP (3rd Generation Partnership Project) ที่มีการพัฒนาให้ LTE มีความเร็วมากกว่ายุค 3G ถึง 10 เท่า โดยมีความสามารถในการส่งถ่ายข้อมูลและมัลติมีเดียสตรีมมิ่งที่มีความเร็วอย่างน้อย 100 Mbps และมีความเร็วสูงสุดถึง 1 Gbps

- ❖ eNodeB (evolved Node B) : รวมเอาหน้าที่ของ RNC เข้าไปด้วย สามารถที่จะจัดการอะไรๆ ได้ด้วยตัวเอง และโครงสร้างที่แบนราบมากขึ้น สามารถที่จะส่งต่อการเชื่อมต่อไปยังสถานีฐานข้างๆ ได้ด้วย X2 interface ที่เชื่อมระหว่างกัน ทำให้มีค่า Latency ต่ำลง ซึ่งเป็นการเปรียบเทียบโครงสร้าง Radio Access Network ของทั้งคู่



ภาพที่ 2.13 การรวมฟังก์ชันเครือข่ายหลักหลายแบบเพื่อประสิทธิภาพที่เหนือกว่า

2.5 Iub Plan

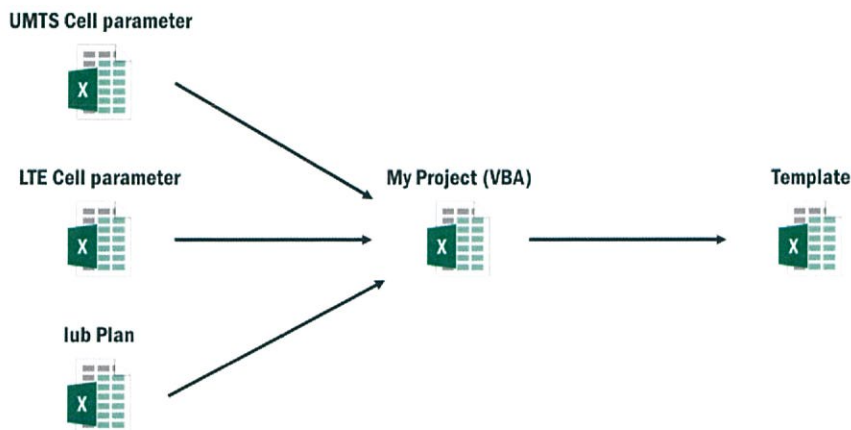
Iub คือ อินเตอร์เฟซระหว่าง NodeB กับ RNC ซึ่ง Iub Plan นั้นรวมรวบข้อมูลสถานะฐาน UMTS ของเน็ตเวิร์ค AIS ในกรุงเทพฯ ทั้งหมด

บทที่ 3

วิธีดำเนินการ

สำหรับวิธีการดำเนินงานในการสร้างเครื่องมือเตรียมข้อมูลในการกำหนดค่าสถานีฐานระบบ 3G และ 4G ต้องเขียนแยกตามประเภทของ BSC ซึ่งจะแบ่งเป็น BSC6900 และ BSC6910 3 ประเภท ได้แก่ 2IPpool, 3IPpool และ 4IPpool รวมแล้วมี 4 ประเภท โดยที่มีวิธีการเขียนโค้ดที่คล้ายคลึงกัน

3.1 การทำงานของระบบ



ภาพที่ 3.1 บล็อกไดอะแกรมของเครื่องมือเตรียมข้อมูลในการกำหนดค่าสถานีฐานระบบ 3G และ 4G

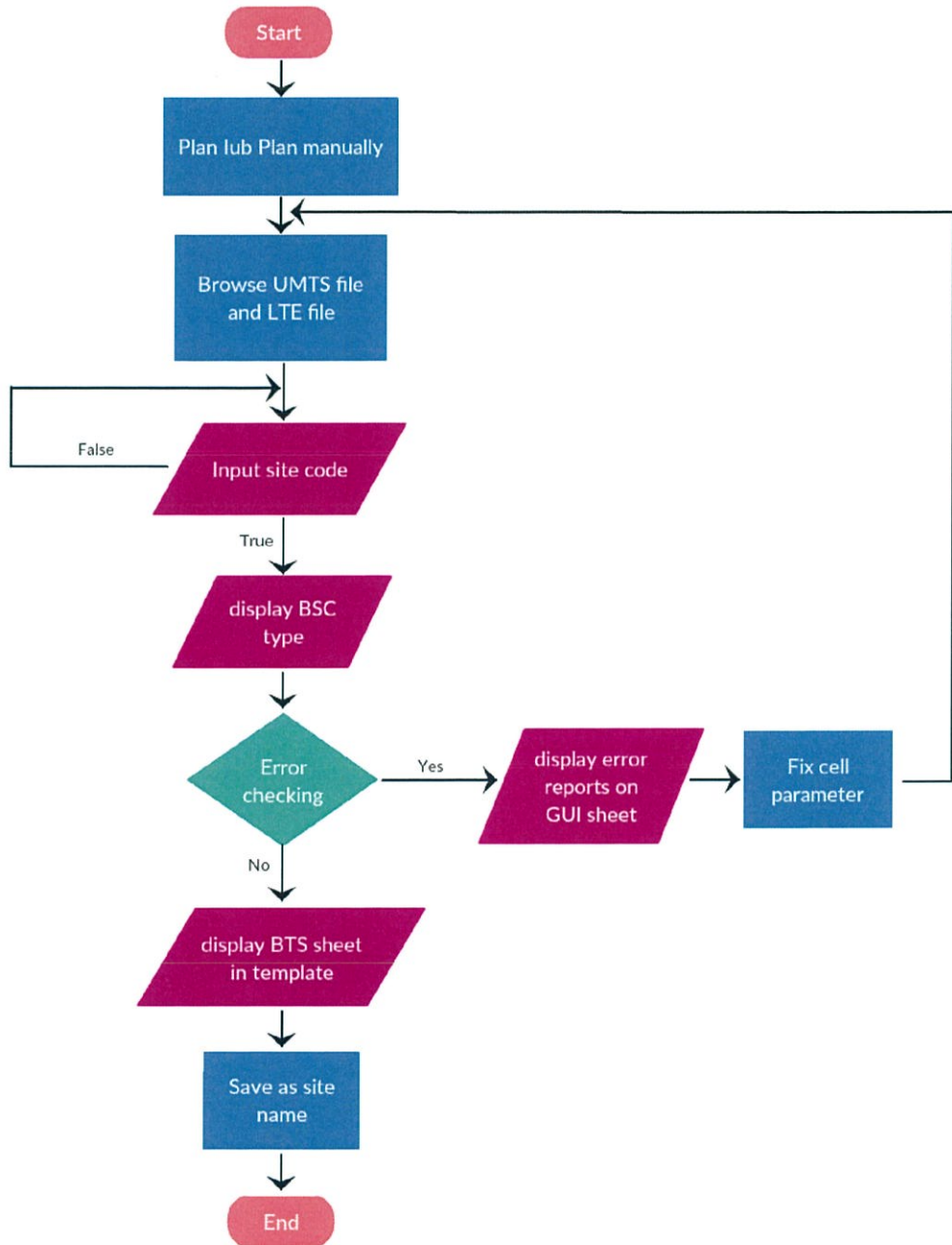
จากบล็อกไดอะแกรมจะมีการทำงานจากไฟล์ excel 3 ไฟล์ ได้แก่ ไฟล์ cell parameter ของ UMTS, ไฟล์ cell parameter ของ LTE และไฟล์ lub Plan โดยกรอกข้อมูลลงในไฟล์ lub Plan ก่อน หลังจากนั้นทำการเบราวส์ไฟล์ cell parameter ของ UMTS และไฟล์ cell parameter ของ LTE ลงในโครงการนี้ ทำการเจนเนอเรทแล้วจะได้เป็นไฟล์ template สำหรับ import ใน CME

3.2 การออกแบบระบบ

การออกแบบเครื่องมือเตรียมข้อมูลในการกำหนดค่าสถานีฐานระบบ 3G และ 4G มีการทำงานเริ่มจากการเขียนโปรแกรมคำสั่งสำหรับการเบราวส์ไฟล์ cell parameter ของ UMTS และไฟล์ cell parameter ของ LTE เมื่อใส่ชื่อสถานีฐานแล้วโปรแกรมจะทำการประมวลผลและแสดงประเภทของ BSC ในหน้าต่าง จากนั้นทำการตรวจสอบข้อผิดพลาด ถ้ามีข้อผิดพลาดจะแสดงข้อมูลที่ผิดพลาดในหน้า GUI

และต้องทำการแก้ไขก่อนจึงจะเริ่มเบราวซ์ไฟล์ใหม่ แต่ถ้าไม่มีข้อผิดพลาด โปรแกรมจะแสดงชื่อ Base Station Transport Data ของไฟล์ template และจะบันทึกไฟล์อัตโนมัติด้วยชื่อสถานีฐาน

แผนผังรูปแบบการทำงาน



ภาพที่ 3.2 Flow chart ของเครื่องมือเตรียมข้อมูลในการกำหนดค่าสถานีฐานระบบ 3G และ 4G

บทที่ 4

ผลการวิจัย

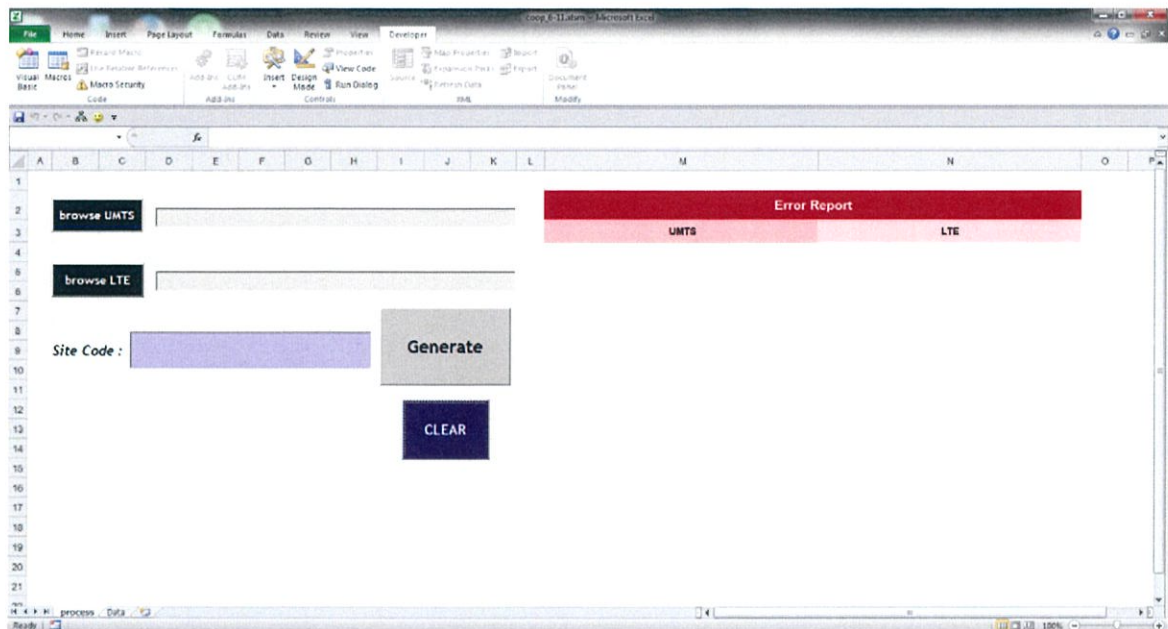
ผลการดำเนินงานในการสร้างเครื่องมือเตรียมข้อมูลในการกำหนดค่าสถานีฐานระบบ 3G และ 4G อยู่ในรูปแบบของ GUI และมีวิธีการใช้งาน ดังนี้

1. กรอกข้อมูลที่ได้รับทางอีเมลล์และจากเซลล์พารามิเตอร์ทั้ง 2 ได้แก่ UMTS(3G) และ LTE (4G) ลงในไฟล์ lub Plan เมื่อกรอกเสร็จให้เซฟและปิดไฟล์นี้

Planner	EIO/AMA FY	Site Code Config	Site Type	RNC (S6 RNCs)	NodeB Name	RRU Type	Part No.	NodeB ID	IP/RAN with DTH Sharing	Chain No STRS	RNC InterworkSubst	
pathamas	10-Aug-2017	UNBCP	Fico	SRNCPRLH	SRNCPRLH_UNBCP	SRU3900	1	877	NO	BKX_C076	0.22.1	
sareerat	11-Aug-2017	PHOCM	Macro	SRNCTAMH	SRNCTWASH_PHOCM	SRU3900	1	710	NO	BKX_C066	0.22.0	
pathamas	11-Aug-2017	PT3FM	Macro	SRNCTAMH	SRNCTWASH_PT3FM	SRU3900	0	734	NO	BKX_S475	0.22.1	
pathamas	11-Aug-2017	TEM96	Fico	SRNCPRLH	SRNCPRLH_TEM96	SRU3900	1	878	NO	BKX_C191	0.22.1	
Nantana	16-Aug-2017	WBCMH	Macro	SRNCTAMH	SRNCTWASH_WBCMH	SRU3900	0	711	NO	BKX_C200	0.22.0	
Nantana	16-Aug-2017	SH14M	Macro	SRNCTAMH	SRNCTWASH_SH14M	SRU3900	0	785	NO	BKX_C016	0.21.0	
Nantana	16-Aug-2017	BAOCM	Macro	SRNCPRLH	SRNCPRLH_BAOCM	SRU3900	1	772	NO	BKX_C024	0.22.1	
pathamas	17-Aug-2017	SH12M_D1	Fico	SRNCPRLH	SRNCPRLH_SH12M_D1	SRU3900	0	722	NO	BKX_C411	1.22.0	
pathamas	17-Aug-2017	SPATM_D2	Fico	SRNCTAMH	SRNCTWASH_SPATM_D2	SRU3900	0	794	NO	BKX_C095	0.20.0	
pathamas	17-Aug-2017	MUAMM_D1	Fico	SRNCPRLH	SRNCPRLH_MUAMM_D1	SRU3900	0	879	NO	BKX_C077	0.22.1	
pathamas	17-Aug-2017	HWFM_D1	Fico	SRNCTAMH	SRNCTWASH_HWFM_D1	SRU3900	0	820	NO	BKX_C180	0.22.0	
Thanasat	20-Aug-2017	R20M	Macro	SRNCPRLH	SRNCPRLH_R20M	SRU3900	0	720	NO	BKX_C178	1.22.0	
Thanasat	20-Aug-2017	SR4M	Macro	SRNCPRLH	SRNCPRLH_SR4M	SRU3900	0	721	NO	BKX_C147	1.22.0	
Thanasat	20-Aug-2017	CR82M	Macro	SRNCPRLH	SRNCPRLH_CR82M	SRU3900	0	728	NO	BKX_C278	0.22.0	
Thanasat	21-Aug-2017	SAHCM	Macro	SRNCPRLH	SRNCPRLH_SAHCM	SRU3900	0	717	NO	BKX_C460	0.22.0	
Nantana	26-Aug-2017	WVCM_D1	Fico	SRNCPRLH	SRNCPRLH_WVCM_D1	SRU3900	0	1244	NO	BKX_C091	0.22.1	
Nantana	26-Aug-2017	VPSMM_D1	Fico	SRNCPRLH	SRNCPRLH_VPSMM_D1	SRU3900	0	1247	NO	BKX_C088	0.22.0	
Thanakorn	23-Aug-2017	TKMP_D1	Fico	SRNCPRLH	SRNCPRLH_TKMP_D1	SRU3900	0	880	NO	BKX_C101	0.22.0	
Nantana	6-Sep-2017	TEM55	Fico	SRNCPRLH	SRNCPRLH_TEM55	SRU3900	1	1526	NO	BKX_C396	0.22.0	
pathamas	7-Sep-2017	CH16M_D2	Fico	SRNCTAMH	SRNCTWASH_CH16M_D2	SRU3900	0	851	NO	BKX_C057	0.22.0	
pathamas	7-Sep-2017	WBRMM_D1	Fico	SRNCPRLH	SRNCPRLH_WBRMM_D1	SRU3900	0	881	NO	BKX_C210	0.22.1	
Nantana	11-Sep-2017	TEM11	Fico	SRNCPRLH	SRNCPRLH_TEM11	SRU3900	1	1587	NO	BKX_C070	0.22.1	
Nantana	11-Sep-2017	TEM99	Fico	SRNCPRLH	SRNCPRLH_TEM99	SRU3900	1	1508	NO	BKX_C080	0.22.1	
7400	Nantana	12-Sep-2017	TUTCM_D5	Macro	SRNCPRLH	SRNCPRLH_TUTCM_D5	SRU3900	0	1509	NO	BKX_C077	0.22.1
7401	Nantana	12-Sep-2017	HWFM_D1	Macro	SRNCPRLH	SRNCPRLH_HWFM_D1	SRU3900	0	1510	NO	BKX_C061	0.22.0
7402	Nantana	12-Sep-2017	HWFM_D1	Macro	SRNCTAMH	SRNCTWASH_HWFM_D1	SRU3900	0	1511	NO	BKX_C200	1.22.0
7403	Nantana	12-Sep-2017	TEM62	Fico	SRNCPRLH	SRNCPRLH_TEM62	SRU3900	0	882	NO	BKX_C155	0.22.1
7404	Nantana	13-Sep-2017	TSUB	Fico	SRNCPRLH	SRNCPRLH_TSUB	SRU3900	0	505	NO	BKX_C100	2.24.0
7405	pathamas	13-Sep-2017	TEM97	Fico	SRNCPRLH	SRNCPRLH_TEM97	SRU3900	1	883	NO	BKX_C074	0.22.1
7406	Nantana	14-Sep-2017	SCT1P_2	Fico	SRNCPRLH	SRNCPRLH_SCT1P_2	SRU3900	0	340	NO	BKX_C408	1.22.0
7407	pathamas	15-Sep-2017	TR696	Fico	SRNCPRLH	SRNCPRLH_TR696	SRU3900	1	729	NO	BKX_C253	0.22.1
7408	Nantana	15-Sep-2017	AP44M	Macro	SRNCTAMH	SRNCTWASH_AP44M	SRU3900	0	786	NO	BKX_C295	0.22.0
7409	Nantana	16-Sep-2017	HWFM_D1	Macro	SRNCPRLH	SRNCPRLH_HWFM_D1	SRU3900	0	100	NO	BKX_C482	2.24.0
7410	Nantana	16-Sep-2017	HWFM_D1	Macro	SRNCTAMH	SRNCTWASH_HWFM_D1	SRU3900	0	717	NO	BKX_C708	0.21.0

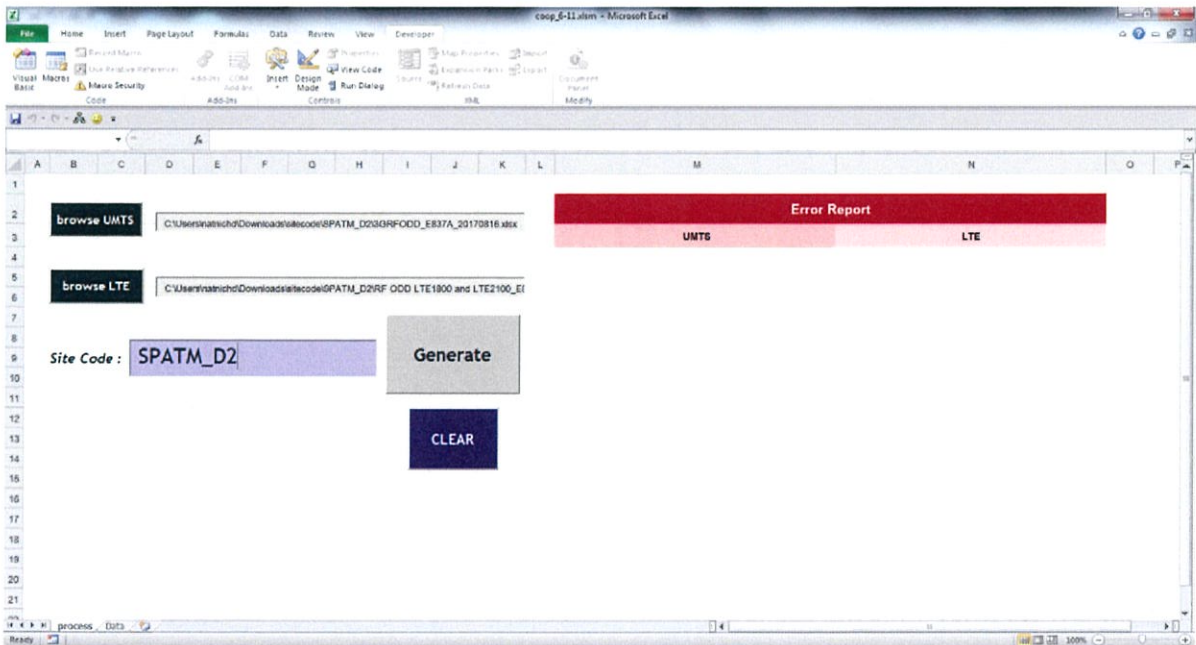
ภาพที่ 4.1 ข้อมูลภายในไฟล์ lub Plan

2. หน้าต่าง GUI (Graphical User Interface) ของโปรแกรม



ภาพที่ 4.2 หน้าต่าง GUI

3. คลิกที่ปุ่ม browse UMTS เลือกไฟล์เซลล์พารามิเตอร์ของ UMTS (3G) และคลิกที่ปุ่ม browse LTE เลือกไฟล์เซลล์พารามิเตอร์ของ LTE (4G) จากนั้นใส่ชื่อสถานีฐานให้ตรงกับเซลล์พารามิเตอร์ทั้ง 2 ในช่องว่าง และคลิกที่ปุ่ม Generate เพื่อสร้างไฟล์ template สำหรับใช้ใน CME



ภาพที่ 4.3 หน้า GUI เมื่อเบรารซ์ไฟล์และใส่ชื่อสถานีฐาน

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	
1	Per Vbloop	RNC Name	NodeB Name (Codeconf)	CellName	Cell ID	Co-Operator Group Index	Band	Bandwidth	UARFCN	Downlink UARFCN	DL Primary Scrambling Code	Location Area Code	Service Area Code	Routing Area Code	Service Priority Group Index	URA E1	URA E2	URA E3	Cell Offset	Time Offset	Local Cell ID	Max Trans Power of
2		3RNC1WA2H	SPATM_02	ER37A0211A	42985		Band1	9612	19762	342	18025	42954	2	1	36008			0	CHP8	31		
3		3RNC1WA2H	SPATM_02	ER37A0211B	42984		Band1	9612	19762	342	18025	42954	2	1	36008			0	CHP256	32		
4		3RNC1WA2H	SPATM_02	ER37A0211C	42988		Band1	9612	19762	326	18025	42958	2	1	36008			0	CHP512	33		
5		3RNC1WA2H	SPATM_02	ER37A0211A	42981		Band1	9636	19766	325	18021	42961	2	1	36008			0	CHP8	41		
6		3RNC1WA2H	SPATM_02	ER37A0214B	42985		Band1	9638	19768	342	18021	42965	2	1	36008			0	CHP256	42		
7		3RNC1WA2H	SPATM_02	ER37A0214C	42989		Band1	9638	19768	348	18021	42969	2	1	36008			0	CHP512	43		
8		3RNC1WA2H	SPATM_02	ER37A0218A	42982		Band1	9662	19812	326	18021	42962	2	1	36008			0	CHP8	51		
9		3RNC1WA2H	SPATM_02	ER37A0218B	42986		Band1	9662	19812	342	18021	42966	2	1	36008			0	CHP256	52		
10		3RNC1WA2H	SPATM_02	ER37A0218C	42976		Band1	9662	19812	326	18021	42976	2	1	36008			0	CHP512	53		
11		3RNC1WA2H	SPATM_02	ER37A0218A	42983		Band1	9687	19817	326	18021	42983	2	1	36008			0	CHP8	61		
12		3RNC1WA2H	SPATM_02	ER37A0218B	42987		Band1	9687	19817	342	18021	42987	2	1	36008			0	CHP256	62		
13		3RNC1WA2H	SPATM_02	ER37A0218C	42971		Band1	9687	19817	348	18021	42971	2	1	36008			0	CHP512	63		

ภาพที่ 4.4 ข้อมูลในไฟล์ UMTS (3G) cell parameter

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U
1	Planner	AIS	AIS	AIS	AIS	AIS	AIS	AIS	AIS	AIS	AIS	Radius	Radius	Radius	Radius	Radius	Radius	Radius	Radius	Radius	Radius
2	Remark	Sitecode	Sitename	Cellcode	CellName	LocalCellID	CellID	SectorID	TAC(Dec)	LAI(LTE2100)	PCI	SSBIndex	SSBRate	PDSCHIndex	PDSCHRate	RSRP	Radius(m)	Emission(B)	PowerSet		
4		ER37A	ja-CHP256	SPATM_02	665091	ER37A5181A	101	101	A	905	19025	235	non-SFN	Null	No-1Ant		408	9000	13.4	Power Set#4	
5		ER37A	ja-CHP256	SPATM_02	665091	ER37A5181B	102	102	B	905	19025	81	non-SFN	Null	No-1Ant		264	9000	13.4	Power Set#4	
6		ER37A	ja-CHP256	SPATM_02	665091	ER37A5181C	103	103	C	905	19025	245	non-SFN	Null	No-1Ant		162	9000	13.4	Power Set#4	

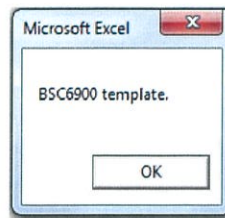
ภาพที่ 4.5 ข้อมูลในไฟล์ LTE (4G) cell parameter

4. เมื่อคลิกปุ่ม Generate แล้ว โปรแกรม Excel จะแสดงหน้าต่างที่มีข้อความว่า “Comfirm to generate?”



ภาพที่ 4.6 หน้าต่างแสดงข้อความว่า “Comfirm to generate?”

5. หลังจากคอนเฟิร์มแล้วรอประมาณ 10 วินาที โปรแกรมจะแสดงประเภทของ BSC



ภาพที่ 4.7 หน้าต่างแสดงประเภทของ BSC

6. ถ้าข้อมูลทั้งหมดถูกต้อง ไฟล์ template จะแสดงขึ้นมาแทนที่หน้าต่าง GUI และจะเซฟไฟล์นี้ไว้ในโฟลเดอร์ NewSiteGenerator ที่กำหนดไว้โดยอัตโนมัติ โดยไฟล์ template ที่ได้นั้นจะมีข้อมูลอยู่ใน worksheet ต่าง ๆ ดังนี้

Base Station		NodeB		Subtrack		ETH Port	Logical NodeB						
Name	Product Type	NodeB Name	UMTS Radio Template	Cabinet No.	Subtrack No.	Subtrack Type	Port No.	RNC Name	NodeB ID	UB Transport Bearer Type	IP Transport Apert Ind	NodeB Host Type	
3R1CTWA2H_SPATM_Q	DDG3600	Co-MPT_3Sector_R19_120	3R1CTWA2H_SPATM_Q	Radio_R19_SPC120	0	0	BDU3900	0	3R1CTWA2H	795	IP_TRAN5	NOT_SUPPORT	SINGLEHOST

ภาพที่ 4.8 Worksheet 1 : Base Station Transport Data

NodeB	Local Cell ID	Local Cell Radius	UT Baseband Equipment ID	DL Baseband Equipment ID	Local Cell Type	UL Frequency Indicator	UL Frequency Channel Number	DL Frequency Channel Number	Max Output Power (dbm)	DL 64QAM
31	29000	0	0	0	NORMAL CELL	TRUE	9812	10762	430	TRUE
32	29000	0	0	0	NORMAL CELL	TRUE	9812	10762	430	TRUE
33	29000	0	0	0	NORMAL CELL	TRUE	9812	10762	430	TRUE
41	29000	0	0	0	NORMAL CELL	TRUE	9838	10788	430	TRUE
42	29000	0	0	0	NORMAL CELL	TRUE	9838	10788	430	TRUE
43	29000	0	0	0	NORMAL CELL	TRUE	9838	10788	430	TRUE
51	29000	1	0	0	NORMAL CELL	TRUE	9862	10812	430	TRUE
52	29000	1	0	0	NORMAL CELL	TRUE	9862	10812	430	TRUE
53	29000	1	0	0	NORMAL CELL	TRUE	9862	10812	430	TRUE
61	29000	1	0	0	NORMAL CELL	TRUE	9807	10837	430	TRUE
62	29000	1	0	0	NORMAL CELL	TRUE	9807	10837	430	TRUE
63	29000	1	0	0	NORMAL CELL	TRUE	9807	10837	430	TRUE

ภาพที่ 4.9 Worksheet 2 : UMTS Cell

NodeB	Local Cell ID	Cell Name	Cell ID	Tracking Area	Cell reserved for operator	Cell operation	MME configuration number	First MME code	First MME group identity	Second MME code	Second MME group identity	Third MME code	Third MME group identity
101	E837A5181A	101	0	0	CELL NOT RESERVED FOR OP	0	CELL MME CFG NUM 4	1	32768	2	32768	3	32768
102	E837A5181B	102	0	0	CELL NOT RESERVED FOR OP	0	CELL MME CFG NUM 4	1	32768	2	32768	3	32768
103	E837A5181C	103	0	0	CELL NOT RESERVED FOR OP	0	CELL MME CFG NUM 4	1	32768	2	32768	3	32768
121	E837A5211A	121	0	0	CELL NOT RESERVED FOR OP	0	CELL MME CFG NUM 4	1	32768	2	32768	3	32768
122	E837A5211B	122	0	0	CELL NOT RESERVED FOR OP	0	CELL MME CFG NUM 4	1	32768	2	32768	3	32768
123	E837A5211C	123	0	0	CELL NOT RESERVED FOR OP	0	CELL MME CFG NUM 4	1	32768	2	32768	3	32768

ภาพที่ 4.10 Worksheet 3 : LTE Cell

NodeB Name	Multiple Carrier Cell Group ID	Multiple Carrier Cell Group Type	Local Cell Ref
SPATM_02	20	HSDPA	4151.61
SPATM_02	21	HSDPA	4252.62
SPATM_02	22	HSDPA	4353.63

ภาพที่ 4.11 Worksheet 4 : Multicarrier Cell Group

NodeB	Local Cell ID	InterRAT handover to UTRAN Group ID	Coverage based UTRAN ECGI trigger threshold (dB)	Coverage based UTRAN RSCP trigger threshold (dBm)	UTRAN handover hysteresis (dB)	UTRAN time to trigger (ms)	Load Service (dB)
SPATM_02	191	0	-28	-103	2	320ms	
SPATM_02	192	0	-28	-103	2	320ms	
SPATM_02	193	0	-28	-103	2	320ms	
SPATM_02	121	0	-28	-103	2	320ms	
SPATM_02	122	0	-28	-103	2	320ms	
SPATM_02	123	0	-28	-103	2	320ms	
SPATM_02	191	1	-28	-103	2	320ms	
SPATM_02	192	1	-28	-103	2	320ms	
SPATM_02	193	1	-28	-103	2	320ms	
SPATM_02	121	1	-28	-103	2	320ms	
SPATM_02	122	1	-28	-103	2	320ms	
SPATM_02	123	1	-28	-103	2	320ms	

ภาพที่ 4.12 Worksheet 5 : InterRatHoUtranGroup

eNodeB	Local cell ID	InterRAT handover common group ID	InterRAT A1A2 hysteresis(0.5dB)	InterRAT A1A2 time to trigger(ms)	InterRAT A1 RSRP trigger threshold(0dBm)	InterRAT A1 RSRQ trigger threshold(0.5dB)	InterRAT A2 RSRP trigger threshold
SPATM D2	181	0	2	640ms	-111	-20	-115
SPATM D2	182	0	2	640ms	-111	-20	-115
SPATM D2	183	0	2	640ms	-111	-20	-115
SPATM D2	121	0	2	640ms	-111	-20	-115
SPATM D2	122	0	2	640ms	-111	-20	-115
SPATM D2	123	0	2	640ms	-111	-20	-115
SPATM D2	181	1	2	640ms	-111	-20	-115
SPATM D2	182	1	2	640ms	-111	-20	-115
SPATM D2	183	1	2	640ms	-111	-20	-115
SPATM D2	121	1	2	640ms	-111	-20	-115
SPATM D2	122	1	2	640ms	-111	-20	-115
SPATM D2	123	1	2	640ms	-111	-20	-115

ภาพที่ 4.13 Worksheet 6 : InterRatHoCommGroup

eNodeB	Local cell ID	Downlink UARF CH	PS service priority	SRVCC Service Priority	Reselection priority config indicator	Cell reselection priority	Frequency Priority for Connected Mode	UTRAN high priority threshold(2dB)	UTRAN lower priority threshold
SPATM D2	181	19762	Priority_2	Priority_16	CFG	3	3	6	6
SPATM D2	182	19762	Priority_2	Priority_16	CFG	3	3	6	6
SPATM D2	183	19762	Priority_2	Priority_16	CFG	3	3	6	6
SPATM D2	121	19762	Priority_2	Priority_16	CFG	3	3	6	6
SPATM D2	122	19762	Priority_2	Priority_16	CFG	3	3	6	6
SPATM D2	123	19762	Priority_2	Priority_16	CFG	3	3	6	6
SPATM D2	181	18789	Priority_16	Priority_2	CFG	3	3	6	6
SPATM D2	182	18789	Priority_16	Priority_2	CFG	3	3	6	6
SPATM D2	183	18789	Priority_16	Priority_2	CFG	3	3	6	6
SPATM D2	121	18789	Priority_16	Priority_2	CFG	3	3	6	6
SPATM D2	122	18789	Priority_16	Priority_2	CFG	3	3	6	6
SPATM D2	123	18789	Priority_16	Priority_2	CFG	3	3	6	6
SPATM D2	181	16812	Priority_16	Priority_2	CFG	3	3	6	6
SPATM D2	182	16812	Priority_16	Priority_2	CFG	3	3	6	6
SPATM D2	183	16812	Priority_16	Priority_2	CFG	3	3	6	6
SPATM D2	121	16812	Priority_16	Priority_2	CFG	3	3	6	6
SPATM D2	122	16812	Priority_16	Priority_2	CFG	3	3	6	6
SPATM D2	123	16812	Priority_16	Priority_2	CFG	3	3	6	6
SPATM D2	181	16837	Priority_16	Priority_2	CFG	3	3	6	6
SPATM D2	182	16837	Priority_16	Priority_2	CFG	3	3	6	6
SPATM D2	183	16837	Priority_16	Priority_2	CFG	3	3	6	6
SPATM D2	121	16837	Priority_16	Priority_2	CFG	3	3	6	6
SPATM D2	122	16837	Priority_16	Priority_2	CFG	3	3	6	6
SPATM D2	123	16837	Priority_16	Priority_2	CFG	3	3	6	6

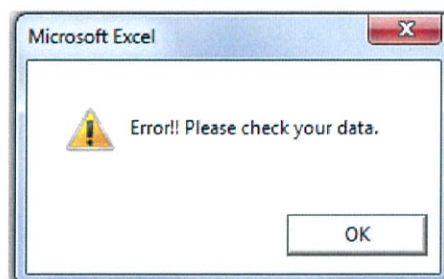
ภาพที่ 4.14 Worksheet 7 : UtranNFreq

The screenshot shows a Microsoft Excel window with the following data in the 'TcelpMapping' worksheet:

eNodeB	TCE ID	Mode	TCE IP Address
SPATM_D2	0	IPv4	10.216.197.157
SPATM_D2	1	IPv4	10.216.197.158
SPATM_D2	2	IPv4	10.216.197.159
SPATM_D2	3	IPv4	10.216.197.160
SPATM_D2	4	IPv4	10.216.197.161
SPATM_D2	5	IPv4	10.216.197.162
SPATM_D2	6	IPv4	10.216.197.163
SPATM_D2	7	IPv4	10.216.197.164
SPATM_D2	8	IPv4	10.216.197.165
SPATM_D2	9	IPv4	10.216.197.166
SPATM_D2	10	IPv4	10.216.197.167
SPATM_D2	11	IPv4	10.216.197.168
SPATM_D2	12	IPv4	10.216.197.169

ภาพที่ 4.15 Worksheet 8 : TcelpMapping

7. กรณีที่เซลล์พารามิเตอร์มีข้อมูลที่ผิดพลาดนั้น โปรแกรม Excel จะแสดงข้อความเตือน “Error!! Please check your data.” เพื่อให้ตรวจสอบข้อมูลในเซลล์พารามิเตอร์



ภาพที่ 4.16 หน้าต่างแสดงข้อความเตือน “Error!! Please check your data.”

8. ในหน้าต่าง GUI จะมีรายงานความผิดพลาด โดยแยกข้อมูลตามเซลล์พารามิเตอร์ของ UMTS และ LTE

Error Report	
UMTS	LTE
Time Offset[chip] = CHIP500 : is invalid.	Cell Name = E837AS181A24 : is invalid.
Time Offset[chip] = CHIP500 : is invalid.	RootSequenceldx = 1000 : Range 0-837 only.
Time Offset[chip] = CHIP500 : is invalid.	RootSequenceldx = 1000 : Range 0-837 only.
Time Offset[chip] = CHIP500 : is invalid.	Local cell = 101 are repeated.
Cell ID = 42964 are repeated values @ cellname : E837AD213A	Cell ID = 101 are repeated values @ cellname : E837AS181B
Cell ID = 42964 are repeated values @ cellname : E837AD216C	Cell ID = 101 are repeated values @ cellname : E837AS181C
Cell ID = 42970 are repeated values @ cellname : E837AD213B	PCI = 1 are repeated values @ cellname : E837AS181A24
Cell ID = 42970 are repeated values @ cellname : E837AD213C	PCI = 1 are repeated values @ cellname : E837AS181B
Cell ID = 42970 are repeated values @ cellname : E837AD213A	RSI = 1000 are repeated values @ cellname : E837AS181B
Cell ID = 42970 are repeated values @ cellname : E837AD214B	RSI = 1000 are repeated values @ cellname : E837AS181C
Cell ID = 42970 are repeated values @ cellname : E837AD214C	
Cell ID = 42970 are repeated values @ cellname : E837AD213A	
Cell ID = 42970 are repeated values @ cellname : E837AD215B	
Cell ID = 42970 are repeated values @ cellname : E837AD215C	
Cell Name = E837AD213A are repeated.	
PSC of sector = E837A-B are 3 different values.	

ภาพที่ 4.17 Error Report

ซึ่งผู้ใช้งานสามารถตรวจสอบข้อมูลได้ในเซลล์พารามิเตอร์ทั้ง 2 โดยที่ในเซลล์พารามิเตอร์ของ UMTS มีรายงานข้อผิดพลาดดังนี้

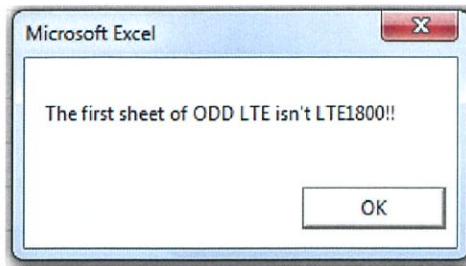
- Time Offset[chip] = CHIP500 ไม่ถูกต้อง เนื่องจาก Time Offset[chip] ต้องมีค่า CHIP0 , CHIP256, CHIP512, CHIP768, CHIP1024, CHIP1280, CHIP1536, CHIP1792, CHIP2048,CHIP2304 เท่านั้น
- Cell ID = 42964 มีค่าซ้ำกันใน Cell name E837AD213A และ E837AD216C
- Cell ID = 42970 มีค่าซ้ำกันใน Cell name : E837AD213B, E837AD213C, E837AD213A, E837AD214B, E837AD214C, E837AD213A, E837AD215B และ E837AD215C
- Cell Name = E837AD213A มีค่าซ้ำกัน
- PSC of sector : E837A-B มีค่าที่ต่างกัน 3 ค่า

Cell ID	Band	Local Cell	Max Trans Power of C
42964	0	0	31
42970	0	0	32
42970	0	0	33
42970	2	0	41
42970	2	0	42
42970	2	0	43
42970	2	0	51
42970	2	0	52
42970	2	0	53
42967	2	0	61
42967	2	0	62
42964	2	0	63

ภาพที่ 4.18 ข้อมูลที่ผิดพลาดในไฟล์ UMTS cell parameter

ส่วนเซลล์พารามิเตอร์ของ LTE มีรายงานข้อผิดพลาดดังนี้

- Cell Name = E837AS181A24 ไม่ถูกต้อง เนื่องจาก Cell Name มีตัวเลขและตัวอักษรรวมกัน 10 ตัวเท่านั้น
- RootSequenceldx = 1000 ทั้ง 2 ค่า ต้องอยู่ในช่วง 0 – 837 เท่านั้น
- Local cell = 101 มีค่าซ้ำกัน
- Cell ID = 101 มีค่าซ้ำกันที่ Cell name : E837AS181B และ E837AS181C
- PCI = 1 มีค่าซ้ำกันที่ Cell name : E837AS181A24 และ E837AS181B
- SI = 1000 มีค่าซ้ำกันที่ Cell name : E837AS181B และ E837AS181C



ภาพที่ 4.21 หน้าต่างแสดงข้อความเตือน “The first sheet of ODD isn’t LTE1800!!”

มีข้อมูลบางอย่างอาจมีการเปลี่ยนแปลงได้ จึงเขียนลงในเซลล์ของ worksheet ที่มีชื่อว่า “Data” ให้ง่ายต่อการแก้ไขเป็นต้น

	A	B
1	LTE Cell Template	
2	LTE1800 Cell Template	LTE_Cell_FDD_15M_AIS_SRAN12_V9
3	LTE2100 Cell Template	LTE_Cell_FDD_10M_AIS_SRAN12_L2100_V1
4	LTE900 Cell Template	LTE_Cell_FDD_5M_AIS_SRAN12_L900_Pico_V1
5		
6	UMTS Cell Template	
7	*Local Cell Template	AIS_LOCELL_R19_SPC120
8		
9	Folder name	ex. : \NewSite ULMode Template_R19SPC120_Rev2.4\
10		\NewSite ULMode Template_R19SPC120_Rev2.4\
11		
12	lub Plan name	ex. : \New AIS 3G2100 NodeB lub Plan update V16.0_20170928.xlsx
13		\New AIS 3G2100 NodeB lub Plan update V16.0_20170928.xlsx
14		
15		
16		
17		
18		
19		
20		
21		
22		

process **Data**

ภาพที่ 4.22 ข้อมูลในชีท Data

สรุปผลการวิจัยและข้อเสนอแนะ

5.1 สรุปผล

โครงการเรื่อง “เครื่องมือเตรียมข้อมูลในการกำหนดค่าสถานีฐานระบบ 3G และ 4G” ที่ได้จัดทำขึ้นในโครงการสหกิจศึกษา เริ่มจากการศึกษาข้อมูลการทำงานของเครื่องเตรียมข้อมูลสำหรับสถานีฐานใหม่แบบเดิม ข้อบกพร่อง และวิเคราะห์ปัญหาการทำงานรูปแบบเดิม ต่อจากนั้นทำการออกแบบกระบวนการทำงานรูปแบบใหม่ และส่วนติดต่อกับผู้ใช้งาน หลังจากนั้นจึงเริ่มการเขียนโปรแกรมและมีการตรวจสอบแก้ไขโปรแกรมที่จัดทำขึ้นเป็นระยะ เพื่อให้ได้โปรแกรมที่มีประสิทธิภาพและเหมาะสมกับการใช้งานมากที่สุด

สำหรับเครื่องมือเตรียมข้อมูลในการกำหนดค่าสถานีฐานระบบ 3G และ 4G ใช้งานได้สำหรับเครือข่ายของ Huawei เพียงเครือข่ายเดียว และจำกัดพื้นที่ในการติดตั้งสถานีฐานใหม่ของกรุงเทพฯ และปริมณฑลเท่านั้น จากการจัดทำโครงการนี้ ได้ใช้ประโยชน์จากการเขียนโปรแกรม VBA (Visual Basic for Application) ใน Microsoft Excel ให้ทำงานแบบอัตโนมัติ เพื่อปรับปรุงโปรแกรมให้มีความสอดคล้องกับขั้นตอนการดำเนินงานในการใช้งานจริงที่ตรงตามความต้องการของผู้ใช้งาน เพื่อช่วยให้ผู้ใช้งานได้รับความสะดวก ลดขั้นตอนของกระบวนการเตรียมข้อมูลในการขึ้นสถานีฐานใหม่ให้มีความรวดเร็วยิ่งขึ้น และลดความผิดพลาดที่อาจเกิดขึ้นได้ อีกทั้งยังประหยัดเวลาในการทำงาน โดยที่มีข้อจำกัดสำหรับการใช้งาน ดังนี้

1. ไฟล์ Cell parameter ของ UMTS ต้องมี header อยู่บรรทัดที่ 1 และไม่มีช่องว่าง
2. ไฟล์ Cell parameter ของ LTE ต้องมี header อยู่บรรทัดที่ 4 และไม่มีช่องว่าง
3. ไฟล์ Cell parameter ของ LTE ต้องมีเลข 18 ประกอบอยู่ในชื่อของซีทแรก
4. หากมีข้อมูลผิดใน Cell parameter จะไม่แสดงไฟล์ template ที่ผิดพลาด
5. เมื่อแสดงไฟล์ template ต้องตรวจสอบข้อมูลทุกครั้งก่อนนำไปใช้

5.2 ปัญหาและอุปสรรค

1. เนื่องจากผู้จัดทำไม่เคยเขียน VBA จึงทำให้ส่วนของการศึกษาการใช้โปรแกรมใช้เวลานาน
2. ในการเขียนโปรแกรม บางครั้งเกิดข้อผิดพลาด (Error) จึงต้องใช้เวลาในการตรวจสอบ

ข้อผิดพลาด และแก้ไขให้ถูกต้อง ทำให้เกิดความล่าช้า

5.3 ข้อเสนอแนะ

เครื่องมือเตรียมข้อมูลในการกำหนดค่าสถานีฐานระบบ 3G และ 4G สามารถที่จะนำไปพัฒนาและปรับปรุงคุณภาพให้ดียิ่งขึ้น หากผู้จัดทำมีความเชี่ยวชาญในการเขียนโปรแกรม VBA จะทำให้การเขียนโปรแกรมสั้นกะทัดรัด และแก้ไขให้โปรแกรมสามารถทำงานได้โดยไม่มีข้อจำกัดข้างต้น เพื่อให้โปรแกรมประมวลผลและทำงานได้อย่างมีประสิทธิภาพมากขึ้น นอกจากนี้ยังสามารถใช้โปรแกรมอื่น เช่น python หรือ java มาใช้ในการเขียนโปรแกรมได้อีกด้วย

เอกสารอ้างอิง

- [1] จีราวุธ วารินทร์. สร้างระบบงานเพื่อจัดการข้อมูลด้วย Excel VBA ฉบับสมบูรณ์. กรุงเทพฯ. : รีไวว่า, 2560
- [2] “Microsoft Excel”. [Online]. <http://www.inwexcel.com>
- [3] “Microsoft Excel Macro คืออะไร”. [Online]. <http://www.9experttraining.com>
- [4] “ข้อมูลและตัวแปร”. [Online]. <http://www.elecnet.chandra.ac.th>
- [5] “ตัวแปรและชนิดของข้อมูล”. [Online]. <http://mpa-r1.blogspot.com/2011/05/3-variable-declaration-variable.html>
- [6] “UMTS Interfaces and Protocols”. [Online]. <http://what-when-how.com/roaming-in-wireless-networks/umts-interfaces-and-protocols-third-generation-networks/>
- [7] “UMTS – 3G Mobile Network”. [Online]. <http://thebestwirelessinternet.com/umts.html>
- [8] “LTE Network Interfaces”. [Online]. <http://4gco.blogspot.com/2017/06/>
- [9] “Evolving to LTE”. [Online]. https://www.cisco.com/c/en/us/solutions/collateral/service-provider/lte-epc/white_paper_c11-609205.html
- [10] “LTE & UMTS Topology”. [Online]. <http://y.tt/13011300070th.html>

ภาคผนวก ก

1. Code VBA : Sheet1(process)

```
Private Sub browse1_click()  
    Module1.browse1  
End Sub
```

```
Private Sub browse2_Click()  
    Module1.browse2  
End Sub
```

```
Private Sub Clear_Click()  
    Module1.Clear  
End Sub
```

```
Private Sub textbox_Change()  
    textbox.Value = UCase(textbox.Value)  
End Sub
```

```
Private Sub generate_Click()  
Dim Msg As String  
Dim Ans As Variant  
    Msg = "Confirm to generate?"  
    Ans = MsgBox(Msg, vbYesNo)  
    Select Case Ans  
    Case vbYes  
        Module1.all  
    Case vbNo  
        GoTo Quit:  
    End Select  
Quit:  
End Sub
```

2. Code VBA : Module1

Option Explicit

```
Public coopWB, planwb, UMTSwb, LTEwb As Workbook
Public myfile1, myfile2 As String
Public BSC6900wb, IP2Poolwb, IP3Poolwb, IP4Poolwb As Workbook
Public BSC6900ws1, BSC6900ws2, BSC6900ws3, BSC6900ws4 As Variant
Public BSC6900ws5, BSC6900ws6, BSC6900ws7, BSC6900ws8 As Variant
Public IP2Poolws1, IP2Poolws2, IP2Poolws3, IP2Poolws4 As Variant
Public IP2Poolws5, IP2Poolws6, IP2Poolws7, IP2Poolws8 As Variant
Public IP3Poolws1, IP3Poolws2, IP3Poolws3, IP3Poolws4 As Variant
Public IP3Poolws5, IP3Poolws6, IP3Poolws7, IP3Poolws8 As Variant
Public IP4Poolws1, IP4Poolws2, IP4Poolws3, IP4Poolws4 As Variant
Public IP4Poolws5, IP4Poolws6, IP4Poolws7, IP4Poolws8 As Variant
Public wbheader(100), positionheader(100) As String
Public wbUheader(100), positionUheader(100) As String
Public wbLheader(100), positionLheader(100) As String
Public planwbSheetName(100), UMTSwbSheetName(100), LTEwbSheetName(100) As String
Public allsite(20000), BSCtypePool, inputsitecode As String
Public sitecode As Variant
Public topic(1 To 100), topicCol(1 To 100) As String
Public topicUMTS(1 To 50), topicUCol(1 To 61) As String
Public topicLTE(1 To 30), topicLCol(1 To 50) As String
Public allldigit, allldigit, allldig As String
Public localcellall(0 To 50) As String
Public localcellallnum As Integer
Public alltopic, alltopicUMTS, alltopicLTE As Integer
Public Bposition, Bsheet, Uposition, Usheet, Lposition, Lsheet As Integer
Public x, y, m, n, r, v, w As Integer
Public topiccolumn, L18num As Integer
Public ODDrow_UMTS As Variant
Public ODD_UMTS, ODDrow_LTE18, ODD_L18, countL18, lastrow As Integer
Public rowallLTE, ODD_allLTE, Ltime, lastR, countLtime As Integer
Public ODD_allL, rowallL, Ltime1, lastR1, countLtime1 As Integer
Public ODD_L, rowL, Ltime2, lastR2, countLtime2 As Integer
Public reltime, lastR3, countLtime3, reltime1, lastR4, countLtime4, reltime2 As Integer
Public ODD_reL, rowreL, countreL, nextR, countreL1, reLow As Integer
Public TCE, addTCE As Integer
Public TCEAddress, cutTCE, allTCE As String
Public countA1, countall, freq, SECTOR, NumofUMTS As Integer
Public allcellname, Rightcellname, Leftcellname, cellname As String
Public siterow As Long
```

```

Publictextbox, TextBox1 As Variant
PubliccountUError, countLError As Integer
PublicgroupS, groupSsize As Variant
PublicgroupS1(0To 5), groupS2(0To 5), groupS3(0To 5) As String
PublicgroupS4(0To 5), groupS5(0To 5), groupS6(0To 5) As String
PublicPSC0, PSC1, PSC2, PSC3, PSC4, PSC5 As Variant
PublicPSC0size, PSC1size, PSC2size, PSC3size, PSC4size, PSC5size As Variant
PublicnumberS, numberP0, numberP1, numberP2, numberP3, numberP4, numberP5 As Integer
PublicPSCdup As Integer
Publicband(0To 9), band1(0To 5), band2(0To 5), band3(0To 5) As String
Publicband4(0To 5), band5(0To 5), band6(0To 5) As String
Publicallnum, allN, pc, pc1, pc3 As Integer
PubliccellnameU(0To 30), localUMTS(0To 30), cellidU(0To 30), PSC0_UMTS(0To 30) As Variant
PubliccellLTE18(0To 30), localLTE18(0To 30), cellidLTE18(0To 30), PCI_LTE18(0To 30),
RSI_LTE18(0To 30) As Variant
Publicword, PSCbox(0To 5) As String
PublicnumP, qq1, qq2, qq3, bx1 As Integer

```

```
PublicSuball()
```

```
On Error Resume Next
```

```
Set coopWB= ThisWorkbook
```

```
countUError = 0
```

```
countLError = 0
```

```
allnum = 0
```

```
allN = 0
```

```
bx1 = 0
```

```
numberS = 0
```

```
numberP0 = 0
```

```
numberP1 = 0
```

```
numberP2 = 0
```

```
numberP3 = 0
```

```
numberP4 = 0
```

```
numberP5 = 0
```

```
groupS = Array("", "", "", "", "", "", "", "", "", "", "")
```

```
groupSsize = UBound(groupS)
```

```
PSC0 = Array("", "", "", "", "", "")
```

```
PSC0size = UBound(PSC0)
```

```
PSC1 = Array("", "", "", "", "", "")
```

```
PSC1size = UBound(PSC1)
```

```
PSC2 = Array("", "", "", "", "", "")
PSC2size = UBound(PSC2)
```

```
PSC3 = Array("", "", "", "", "", "")
PSC3size = UBound(PSC3)
```

```
PSC4 = Array("", "", "", "", "", "")
PSC4size = UBound(PSC4)
```

```
PSC5 = Array("", "", "", "", "", "")
PSC5size = UBound(PSC5)
```

```
coopWB.Worksheets("process").Rows("4:" & Rows.Count).ClearContents
coopWB.Worksheets("process").Rows("4:" & Rows.Count).Interior.Color = xlNone
coopWB.Worksheets("process").Cells.HorizontalAlignment = xlCenter
```

```
'-----Base Station Transport Data-----'
```

```
'--- lub plan header = Base header ---'
```

```
topic(1) = "Site Code Config"
topic(2) = "RNC (16 RNCs)"
topic(3) = "NodeB Name"
topic(4) = "BBU Type"
topic(5) = "Port No."
topic(6) = "BSCPool"
topic(7) = "NodeB ID"
topic(8) = "Route Remark"
topic(9) = "NodeB Series"
topic(10) = "NodeB Template"
topic(11) = "Logical RNCID"
topic(12) = "SCTPLINK_ID_NCP"
topic(13) = "SCTPLINK_ID_CCP"
topic(14) = "Device IP"
topic(15) = "Subrack_No."
topic(16) = "Slot_No."
topic(17) = "Service GWIP"
topic(18) = "Service RAG_Master"
topic(19) = "Service RAG_Slave"
topic(20) = "Sevice VLAN"
topic(21) = "Service IP"
topic(22) = "Destination Mask"
topic(23) = "O&M GW IP"
topic(24) = "OAM RAG_Master"
topic(25) = "OAM RAG_Slave"
```

topic(26) = "O&M VLAN"
topic(27) = "O&M IP"
topic(28) = "ETH Network IP"
topic(29) = "IPPool#2"
topic(30) = "IPPool#3"
topic(31) = "IPPool#4"
topic(32) = "IPCLK1 service IP"
topic(33) = "IPCLK1 network IP"
topic(34) = "IPCLK2 Service IP"
topic(35) = "IPCLK2 network IP"
topic(36) = "eNodeB ID"
topic(37) = "LTE IP"
topic(38) = "LTE Subnet Mask"
topic(39) = "LTE Gateway"
topic(40) = "LTE VLAN"
topic(41) = "M2000 IP"
topic(42) = "*NodeB IP_TRANS Subrack No"
topic(43) = "*NodeB IP_TRANS Slot No."
topic(44) = "ETH GW IP"
topic(45) = "Service Network IP"
topic(46) = "M2000 Network IP"
topic(47) = "SCTP(NCP) No."
topic(48) = "SCTP(CCP) No."
topic(49) = "SPU Subsystem"
topic(50) = ""

alltopic= 49

-----UMTS Cell-----

topicUMTS(1) = "RNC Name"
topicUMTS(2) = "NodeB"
topicUMTS(3) = "CellName"
topicUMTS(4) = "Cell ID"
topicUMTS(5) = "Cn Operator Group Index"
topicUMTS(6) = "Band Indicator"
topicUMTS(7) = "Uplink UARFCN"
topicUMTS(8) = "Downlink UARFCN"
topicUMTS(9) = "DL Primary Scrambling Code"
topicUMTS(10) = "Location Area Code"
topicUMTS(11) = "Service Area Code"
topicUMTS(12) = "Routing Area Code"
topicUMTS(13) = "Service Priority Group Identity"
topicUMTS(14) = "URA ID1"
topicUMTS(15) = "Cell Oriented Cell Individual Offset"

topicUMTS(16) = "Time Offset"
topicUMTS(17) = "Local Cell"
topicUMTS(18) = "Max Transmit Power"
topicUMTS(19) = "PCPICH"
topicUMTS(20) = "Carrier Indicator"
topicUMTS(21) = "Sector No"
topicUMTS(22) = "Sector Quantity"
topicUMTS(23) = "11"
topicUMTS(24) = ""
topicUMTS(25) = ""
topicUMTS(26) = ""
topicUMTS(27) = ""
topicUMTS(28) = ""
topicUMTS(29) = ""
topicUMTS(30) = ""

alltopicUMTS = 22

-----LTE Cell-----

topicLTE(1) = "Sitecode"
topicLTE(2) = "eNodeBNameconfig"
topicLTE(3) = "eNodeBID"
topicLTE(4) = "CellName"
topicLTE(5) = "LocalCellIID"
topicLTE(6) = "CellIID"
topicLTE(7) = "SectorID"
topicLTE(8) = "TAC(Dec)"
topicLTE(9) = "LAC(U2100)"
topicLTE(10) = "PCI"
topicLTE(11) = "MinRootSequenceIndex"
topicLTE(12) = "ULEARFCN"
topicLTE(13) = "DLEARFCN"
topicLTE(14) = "Bandwidth"
topicLTE(15) = "Radius(m)"
topicLTE(16) = "RSpower"
topicLTE(17) = ""
topicLTE(18) = ""
topicLTE(19) = ""
topicLTE(20) = ""
topicLTE(21) = ""
topicLTE(22) = ""
topicLTE(23) = ""
topicLTE(24) = ""
topicLTE(25) = ""

alltopicLTE = 16

'----- set -----'

```
If myfile1 <> "False" And myfile1 <> "" And _  
    myfile2 <> "False" And myfile2 <> "" Then
```

```
    Application.ScreenUpdating = False  
    Application.DisplayAlerts = False  
    Application.AskToUpdateLinks = False
```

```
    Dim lubplan As String  
    lubplan = ThisWorkbook.Worksheets("Data").Range("B10")
```

```
    If Right(lubplan, 1) <> "\" Then  
        Set planwb = Workbooks.Open(coopWB.Path & lubplan, password:="", ReadOnly:=True)  
    End If
```

```
    Set UMTSwb = Workbooks.Open(myfile1)  
    Set LTEwb = Workbooks.Open(myfile2)
```

```
    Call findSheetName  
    Call findHeaderName
```

'----- Read the BASE header name -----'

```
x = 1  
Do While x <> alltopic + 1  
    y = 1  
    Do While y <> 100  
        If topic(x) = wbheader(y) Then  
            topicCol(x) = positionheader(y)  
        End If  
        y = y + 1  
    Loop  
    x = x + 1  
Loop
```

'----- Read the UMTS header name -----'

```
m = 1  
Do While m < alltopicUMTS + 1  
    n = 2  
    Do While n < 60
```

```

        If InStr(1, UMTSwb.Worksheets(UMTSwbSheetName(1)).Cells(1, n).Text, topicUMTS(m))
= 1 Then
            topicUCol(m) = Str(n)
            'MsgBox topicUCol(4) & " = " & Str(4)
        End If
        n = n + 1
    Loop
    m = m + 1
Loop

```

----- Read the LTE header name -----

```
Dim LTEheader As String
```

```
v = 1
```

```
Do While v < alltopicLTE + 1
```

```
    w = 2
```

```
    Do While w < 50
```

```
        LTEheader = LTEwb.Worksheets(LTEwbSheetName(1)).Cells(4, w).Value
```

```
        'MsgBox Str(erer)
```

```
        If LTEwbSheetName(1) Like "*18*" Then
```

```
            If Replace(CStr(LTEheader), " ", "") Like "*" & topicLTE(v) & "*" Then
```

```
                topicLCol(v) = Str(w)
```

```
            End If
```

```
        Else
```

```
            MsgBox "The first sheet of ODD LTE isn't LTE1800!!"
```

```
            planwb.Close savechanges:=False
```

```
            UMTSwb.Close savechanges:=False
```

```
            LTEwb.Close savechanges:=False
```

```
            Exit Sub
```

```
        End If
```

```
    w = w + 1
```

```
    Loop
```

```
v = v + 1
```

```
Loop
```

----- Input side code -----

```
inputsitecode = ""
```

```
inputsitecode = ThisWorkbook.Worksheets("process").textbox.Value
```

```
siterow = 0
```

```
r = 3
```

```
Do While planwb.Worksheets(planwbSheetName(1)).Cells(r, Val(topicCol(1))).Value <> ""
allsite(r) = planwb.Worksheets(planwbSheetName(1)).Cells(r, Val(topicCol(1))).Value
```

```
    If inputsitecode = allsite(r) Then
```

```
        siterow = r
```

```
        BSCTypePool = ""
```

```
        BSCTypePool = planwb.Worksheets(planwbSheetName(1)).Cells(r, Val(topicCol(6))).Value
```

```
    End If
```

```
r = r + 1
```

```
Loop
```

```
'----- Condition for select BSCType -----'
```

```
If siterow <> 0 Then '#Found'
```

```
    If BSCTypePool = "NO" Then
```

```
        MsgBox "BSC6900 template."
```

```
        Module2.BSC6900
```

```
        Module6.BSC6900remain
```

```
        planwb.Close savechanges:=False
```

```
        UMTSwb.Close savechanges:=False
```

```
        LTEwb.Close savechanges:=False
```

```
    ElseIf BSCTypePool = "2IPPOOLS" Then
```

```
        MsgBox "2IPPOOLS template."
```

```
        Module3.IP2Pool
```

```
        Module7.IP2Poolremain
```

```
        planwb.Close savechanges:=False
```

```
        UMTSwb.Close savechanges:=False
```

```
        LTEwb.Close savechanges:=False
```

```
    ElseIf BSCTypePool = "3IPPOOLS" Then
```

```
        MsgBox "3IPPOOLS template."'
```

```
        Module4.IP3Pool
```

```
        Module8.IP3Poolremain
```

```
        planwb.Close savechanges:=False
```

```
        UMTSwb.Close savechanges:=False
```

```
        LTEwb.Close savechanges:=False
```

```
    ElseIf BSCTypePool = "4IPPOOLS" Then
```

```
        MsgBox "4IPPOOLS template."'
```

```
        Module5.IP4Pool
```

```
        Module9.IP4Poolremain
```

```
        planwb.Close savechanges:=False
```

```

    UMTSwb.Close savechanges:=False
    LTEwb.Close savechanges:=False

    End If
Else
    MsgBox "Input Site Code not found.", vbExclamation
    planwb.Close savechanges:=False
    UMTSwb.Close savechanges:=False
    LTEwb.Close savechanges:=False
End If

Else
    MsgBox "Cannot process, please browse file first!", vbExclamation
End If

Application.ScreenUpdating = True
Application.DisplayAlerts = True

Resume Next

End Sub
-----
Sub findHeaderName()

    Bposition = 1
    Do While planwb.Worksheets(planwbSheetName(1)).Cells(2, Bposition).Text <> ""
        wbheader(Bposition) = planwb.Worksheets(planwbSheetName(1)).Cells(2, Bposition).Text
        positionheader(Bposition) = Bposition
        Bposition = Bposition + 1
    Loop

End Sub
-----
Sub findSheetName()

    For Bsheet = 1 To planwb.Sheets.Count
        planwbSheetName(Bsheet) = planwb.Worksheets(Bsheet).Name
    Next Bsheet

    For Usheet = 1 To UMTSwb.Sheets.Count
        UMTSwbSheetName(Usheet) = UMTSwb.Worksheets(Usheet).Name
    Next Usheet

    For Lsheet = 1 To LTEwb.Sheets.Count
        LTEwbSheetName(Lsheet) = LTEwb.Worksheets(Lsheet).Name

```

NextLsheet

End Sub

PublicSub browse1()

```
myfile1 = Application.GetOpenFilename(Title:="Please choose a file to open", _  
fileFilter:="Excel Files *.xls*(*.xls*),")  
Worksheets("process").Textbrowse1.Text = myfile1
```

```
If myfile1 <> "False" Then
```

```
    ThisWorkbook.Worksheets("process").Textbrowse1.Text = myfile1
```

```
Exit Sub
```

```
Else
```

```
MsgBox "No file selected", vbExclamation, "Try again!"
```

```
Exit Sub
```

```
End If
```

```
With CreateObject("excel.application")
```

```
    .Visible = True
```

```
    .Workbooks.Open myfile1
```

```
End With
```

```
Exit Sub
```

End Sub

PublicSub browse2()

```
myfile2 = Application.GetOpenFilename(Title:="Please choose a file to open", _  
fileFilter:="Excel Files *.xls*(*.xls*),")  
Worksheets("process").Textbrowse2.Text = myfile2
```

```
If myfile2 <> "False" Then
```

```
    ThisWorkbook.Worksheets("process").Textbrowse2.Text = myfile2
```

```
Exit Sub
```

```
Else
```

```
MsgBox "No file selected", vbExclamation, "Try again!"
```

```
Exit Sub
```

```
End If
```

```
With CreateObject("excel.application")
```

```
    .Visible = True
```

```
    .Workbooks.Open myfile2
```

```
End With
```

```
Exit Sub
```

End Sub

Public Sub renewPSC(PSCdata)

```
PSCbox(0) = "-"
PSCbox(1) = "-"
PSCbox(2) = "-"
PSCbox(3) = "-"
PSCbox(4) = "-"
PSCbox(5) = "-"
```

```
qq1 = 0
qq3 = 0
Do While qq1 < 6
    word = "FALSE"
    qq2 = 0
    MsgBox PSCdata(qq1)
    Do While qq2 < 6
        MsgBox PSCbox(qq2)
        If PSCdata(qq1) = PSCbox(qq2) Then
            word = "TRUE"
        End If
        qq2 = qq2 + 1
    Loop
    MsgBox word
    If word = "FALSE" Then
        PSCbox(qq3) = PSCdata(qq1)
        MsgBox "PSCbox" & "(" & qq3 & ")" & PSCbox(qq3)
        MsgBox "PSCdata" & "(" & qq1 & ")" & PSCdata(qq1)
        qq3 = qq3 + 1
    End If
    qq1 = qq1 + 1
Loop
PSCdup = qq3 - 1
```

End Sub

Function PSCrepeat(Arr As Variant)

```
Dim Arrsize, l, j, k, n As Integer
Arrsize = UBound(Arr)
Dim tmpArr As Variant
ReDim tmpArr(0 To Arrsize)
Dim Flag As String
```

```
For i = 0 To UBound(Arr)
```

```
For j = 0 To UBound(Arr)
    Flag = "N"
    If i = j Then

        ElseIf Arr(i) = Arr(j) Then
            tmpArr(i) = Arr(i)
        End If
    Next j
Next i
```

```
PSCrepeat = tmpArr
```

```
End Function
```

3. Code VBA : Module2

```
Sub BSC6900()
```

```
Dim folderpath, filepath As String
```

```
folderpath = coopWB.Path & ThisWorkbook.Worksheets("Data").Range("B9").Value
```

```
filepath = Dir(folderpath & "*BSC6900*.xlsm")
```

```
If filepath <> "" Then
```

```
    Set BSC6900wb = Workbooks.Open(folderpath & filepath)
```

```
End If
```

```
'----- BSC6900 BASE -----'
```

```
BSC6900ws1 = "Base Station Transport Data"
```

```
BSC6900wb.Worksheets(BSC6900ws1).Rows("3:" & Rows.Count).ClearContents
```

```
BSC6900wb.Worksheets(BSC6900ws1).Rows("3:" & Rows.Count).Interior.Color = xlNone
```

```
BSC6900wb.Worksheets(BSC6900ws1).Cells.HorizontalAlignment = xlCenter
```

```
'set font and size
```

```
With BSC6900wb.Worksheets(BSC6900ws1)
```

```
    .Cells.Font.Name = "Arial"
```

```
    .Cells.Font.Size = 10
```

```
End With
```

```
    topiccolumn = 1
```

```
    Do While topiccolumn <> 60
```

```
        BSC6900wb.Worksheets(BSC6900ws1).Cells(3, topiccolumn) = ""
```

```
        topiccolumn = topiccolumn + 1
```

```
    Loop
```

```
    BSC6900wb.Worksheets(BSC6900ws1).Cells(3, 1) =
```

```
planwb.Worksheets(planwbSheetName(1)).Cells(siterow, Val(topicCol(3))).Value
```

```
    BSC6900wb.Worksheets(BSC6900ws1).Cells(3, 2) =
```

```
planwb.Worksheets(planwbSheetName(1)).Cells(siterow, Val(topicCol(9))).Value
```

```
    BSC6900wb.Worksheets(BSC6900ws1).Cells(3, 3) =
```

```
planwb.Worksheets(planwbSheetName(1)).Cells(siterow, Val(topicCol(10))).Value
```

```
    BSC6900wb.Worksheets(BSC6900ws1).Cells(3, 4) =
```

```
planwb.Worksheets(planwbSheetName(1)).Cells(siterow, Val(topicCol(3))).Value
```

```
    BSC6900wb.Worksheets(BSC6900ws1).Cells(3, 5) = "Radio_R19_SPC120"
```

```
    BSC6900wb.Worksheets(BSC6900ws1).Cells(3, 6) = "0"
```

```
    BSC6900wb.Worksheets(BSC6900ws1).Cells(3, 7) = "0"
```

```

BSC6900wb.Worksheets(BSC6900ws1).Cells(3, 8) =
planwb.Worksheets(planwbSheetName(1)).Cells(siterow, Val(topicCol(4))).Value
BSC6900wb.Worksheets(BSC6900ws1).Cells(3, 9) =
planwb.Worksheets(planwbSheetName(1)).Cells(siterow, Val(topicCol(5))).Value
BSC6900wb.Worksheets(BSC6900ws1).Cells(3, 10) =
planwb.Worksheets(planwbSheetName(1)).Cells(siterow, Val(topicCol(2))).Value
BSC6900wb.Worksheets(BSC6900ws1).Cells(3, 11) =
planwb.Worksheets(planwbSheetName(1)).Cells(siterow, Val(topicCol(7))).Value
BSC6900wb.Worksheets(BSC6900ws1).Cells(3, 12) = "IP_TRANS"
BSC6900wb.Worksheets(BSC6900ws1).Cells(3, 13) = "NOT_SUPPORT"
BSC6900wb.Worksheets(BSC6900ws1).Cells(3, 14) = "SINGLEHOST"
BSC6900wb.Worksheets(BSC6900ws1).Cells(3, 15) = "RANSHARING"
BSC6900wb.Worksheets(BSC6900ws1).Cells(3, 16) =
planwb.Worksheets(planwbSheetName(1)).Cells(siterow, Val(topicCol(15))).Value
BSC6900wb.Worksheets(BSC6900ws1).Cells(3, 17) =
planwb.Worksheets(planwbSheetName(1)).Cells(siterow, Val(topicCol(16))).Value
BSC6900wb.Worksheets(BSC6900ws1).Cells(3, 18) =
planwb.Worksheets(planwbSheetName(1)).Cells(siterow, Val(topicCol(49))).Value
BSC6900wb.Worksheets(BSC6900ws1).Cells(3, 19) =
planwb.Worksheets(planwbSheetName(1)).Cells(siterow, Val(topicCol(15))).Value
BSC6900wb.Worksheets(BSC6900ws1).Cells(3, 20) =
planwb.Worksheets(planwbSheetName(1)).Cells(siterow, Val(topicCol(16))).Value
BSC6900wb.Worksheets(BSC6900ws1).Cells(3, 21) =
planwb.Worksheets(planwbSheetName(1)).Cells(siterow, Val(topicCol(49))).Value
BSC6900wb.Worksheets(BSC6900ws1).Cells(3, 22) =
planwb.Worksheets(planwbSheetName(1)).Cells(siterow, Val(topicCol(47))).Value
BSC6900wb.Worksheets(BSC6900ws1).Cells(3, 23) =
planwb.Worksheets(planwbSheetName(1)).Cells(siterow, Val(topicCol(12))).Value
BSC6900wb.Worksheets(BSC6900ws1).Cells(3, 24) =
planwb.Worksheets(planwbSheetName(1)).Cells(siterow, Val(topicCol(48))).Value
BSC6900wb.Worksheets(BSC6900ws1).Cells(3, 25) =
planwb.Worksheets(planwbSheetName(1)).Cells(siterow, Val(topicCol(13))).Value
BSC6900wb.Worksheets(BSC6900ws1).Cells(3, 26) =
planwb.Worksheets(planwbSheetName(1)).Cells(siterow, Val(topicCol(14))).Value
BSC6900wb.Worksheets(BSC6900ws1).Cells(3, 27) =
planwb.Worksheets(planwbSheetName(1)).Cells(siterow, Val(topicCol(42))).Value
BSC6900wb.Worksheets(BSC6900ws1).Cells(3, 28) =
planwb.Worksheets(planwbSheetName(1)).Cells(siterow, Val(topicCol(43))).Value
BSC6900wb.Worksheets(BSC6900ws1).Cells(3, 29) =
planwb.Worksheets(planwbSheetName(1)).Cells(siterow, Val(topicCol(15))).Value
BSC6900wb.Worksheets(BSC6900ws1).Cells(3, 30) =
planwb.Worksheets(planwbSheetName(1)).Cells(siterow, Val(topicCol(16))).Value
BSC6900wb.Worksheets(BSC6900ws1).Cells(3, 31) =
planwb.Worksheets(planwbSheetName(1)).Cells(siterow, Val(topicCol(45))).Value

```



```

BSC6900wb.Worksheets(BSC6900ws1).Cells(3, 54) =
planwb.Worksheets(planwbSheetName(1)).Cells(siterow, Val(topicCol(1))).Value
BSC6900wb.Worksheets(BSC6900ws1).Cells(3, 55) =
planwb.Worksheets(planwbSheetName(1)).Cells(siterow, Val(topicCol(36))).Value
BSC6900wb.Worksheets(BSC6900ws1).Cells(3, 56) = "LTE_Radio_AIS_SRAN12_V7"
BSC6900wb.Worksheets(BSC6900ws1).Cells(3, 57) =
planwb.Worksheets(planwbSheetName(1)).Cells(siterow, Val(topicCol(37))).Value
BSC6900wb.Worksheets(BSC6900ws1).Cells(3, 58) =
planwb.Worksheets(planwbSheetName(1)).Cells(siterow, Val(topicCol(38))).Value
BSC6900wb.Worksheets(BSC6900ws1).Cells(3, 59) =
planwb.Worksheets(planwbSheetName(1)).Cells(siterow, Val(topicCol(39))).Value
BSC6900wb.Worksheets(BSC6900ws1).Cells(3, 60) =
planwb.Worksheets(planwbSheetName(1)).Cells(siterow, Val(topicCol(40))).Value
BSC6900wb.Worksheets(BSC6900ws1).Cells(3, 61) = "INSTALL"

```

```

With BSC6900wb.Worksheets(BSC6900ws1)
    .Rows("4" & ":" & .Rows.Count).Delete
End With

```

'set gridlines

```

BSC6900wb.Worksheets(BSC6900ws1).Activate
ActiveSheet.UsedRange.Select
Selection.Borders(xlDiagonalDown).LineStyle = xlNone
Selection.Borders(xlDiagonalUp).LineStyle = xlNone
With Selection.Borders(xlEdgeLeft)
    .LineStyle = xlContinuous
    .ColorIndex = 0
    .TintAndShade = 0
    .Weight = xlThin
End With
With Selection.Borders(xlEdgeTop)
    .LineStyle = xlContinuous
    .ColorIndex = 0
    .TintAndShade = 0
    .Weight = xlThin
End With
With Selection.Borders(xlEdgeBottom)
    .LineStyle = xlContinuous
    .ColorIndex = 0
    .TintAndShade = 0
    .Weight = xlThin
End With
With Selection.Borders(xlEdgeRight)
    .LineStyle = xlContinuous
    .ColorIndex = 0

```

```

        .TintAndShade = 0
        .Weight = xlThin
    End With
    With Selection.Borders(xlInsideVertical)
        .LineStyle = xlContinuous
        .ColorIndex = 0
        .TintAndShade = 0
        .Weight = xlThin
    End With
    With Selection.Borders(xlInsideHorizontal)
        .LineStyle = xlContinuous
        .ColorIndex = 0
        .TintAndShade = 0
        .Weight = xlThin
    End With

```

'----- BSC6900 UMTS -----'

```
BSC6900ws2 = "UMTS Cell"
```

```

BSC6900wb.Worksheets(BSC6900ws2).Rows("3:" & Rows.Count).ClearContents
BSC6900wb.Worksheets(BSC6900ws2).Rows("3:" & Rows.Count).Interior.Color = xlNone
BSC6900wb.Worksheets(BSC6900ws2).Cells.HorizontalAlignment = xlCenter

```

```
'set font and size
```

```

With BSC6900wb.Worksheets(BSC6900ws2)
    .Cells.Font.Name = "Arial"
    .Cells.Font.Size = 10
End With

```

```
countU = 0
```

```
ODDrow_UMTS = 2
```

```
ODD_UMTS = 3
```

```
Dim localindex As Integer
```

```
localindex = 0
```

```

Do While UMTSwb.Worksheets(UMTSwbSheetName(1)).Cells(ODDrow_UMTS,
Val(topicUCol(2))) <> ""

```

```

    If UMTSwb.Worksheets(UMTSwbSheetName(1)).Cells(ODDrow_UMTS,
Val(topicUCol(2))).Value = inputsitecode Then
        countU = countU + 1

```

```

        BSC6900wb.Worksheets(BSC6900ws2).Cells(ODD_UMTS, 1) =
planwb.Worksheets(planwbSheetName(1)).Cells(siterow, Val(topicCol(3))).Value

```

```
BSC6900wb.Worksheets(BSC6900ws2).Cells(ODD_UMTS, 2) =  
UMTSwb.Worksheets(UMTSwbSheetName(1)).Cells(ODDrow_UMTS, Val(topicUCol(17))).Value
```

```
'localcellallnum  
If Left(BSC6900wb.Worksheets(BSC6900ws2).Cells(ODD_UMTS, 2), 1) <> "3" Then  
    localcellall(localindex) = BSC6900wb.Worksheets(BSC6900ws2).Cells(ODD_UMTS, 2)  
    'MsgBox localcellall(localindex)  
    localindex = localindex + 1  
    localcellallnum = localindex  
End If
```

```
'localcell  
Dim CheckUMTSlocalcell As String  
CheckUMTSlocalcell = BSC6900wb.Worksheets(BSC6900ws2).Cells(ODD_UMTS, 2).Value  
localUMTS(allnum) = CheckUMTSlocalcell
```

```
If Len(CheckUMTSlocalcell) <> 2 Then  
    coopWB.Worksheets("process").Cells(4 + countUError, 13) = "Local Cell ID = " &  
CheckUMTSlocalcell & ": is invalid."  
    countUError = countUError + 1  
End If
```

```
BSC6900wb.Worksheets(BSC6900ws2).Cells(ODD_UMTS, 3) = "29000"  
BSC6900wb.Worksheets(BSC6900ws2).Cells(ODD_UMTS, 5) = "0"  
BSC6900wb.Worksheets(BSC6900ws2).Cells(ODD_UMTS, 6) = "NORMAL_CELL"  
BSC6900wb.Worksheets(BSC6900ws2).Cells(ODD_UMTS, 7) = "TRUE"
```

```
'-----column 8,9-----'  
'Freq of UL and DL  
alldigit = UMTSwb.Worksheets(UMTSwbSheetName(1)).Cells(ODDrow_UMTS,  
Val(topicUCol(3))).Value
```

```
If Mid(alldigit, 9, 1) = 1 Then 'freg = 1X  
    BSC6900wb.Worksheets(BSC6900ws2).Cells(ODD_UMTS, 8) = "9763"  
    BSC6900wb.Worksheets(BSC6900ws2).Cells(ODD_UMTS, 9) = "10713"  
ElseIf Mid(alldigit, 9, 1) = 2 Then 'freg = 2X  
    BSC6900wb.Worksheets(BSC6900ws2).Cells(ODD_UMTS, 8) = "9787"  
    BSC6900wb.Worksheets(BSC6900ws2).Cells(ODD_UMTS, 9) = "10737"  
ElseIf Mid(alldigit, 9, 1) = 3 Then 'freg = 3X  
    BSC6900wb.Worksheets(BSC6900ws2).Cells(ODD_UMTS, 8) = "9812"  
    BSC6900wb.Worksheets(BSC6900ws2).Cells(ODD_UMTS, 9) = "10762"  
ElseIf Mid(alldigit, 9, 1) = 4 Then 'freg = 4X  
    BSC6900wb.Worksheets(BSC6900ws2).Cells(ODD_UMTS, 8) = "9838"  
    BSC6900wb.Worksheets(BSC6900ws2).Cells(ODD_UMTS, 9) = "10788"  
ElseIf Mid(alldigit, 9, 1) = 5 Then 'freg = 5X
```

```

BSC6900wb.Worksheets(BSC6900ws2).Cells(ODD_UMTS, 8) = "9862"
BSC6900wb.Worksheets(BSC6900ws2).Cells(ODD_UMTS, 9) = "10812"
Else 'freg = 6X
    BSC6900wb.Worksheets(BSC6900ws2).Cells(ODD_UMTS, 8) = "9887"
    BSC6900wb.Worksheets(BSC6900ws2).Cells(ODD_UMTS, 9) = "10837"
End If

BSC6900wb.Worksheets(BSC6900ws2).Cells(ODD_UMTS, 10) = "430"
BSC6900wb.Worksheets(BSC6900ws2).Cells(ODD_UMTS, 11) = "TRUE"
BSC6900wb.Worksheets(BSC6900ws2).Cells(ODD_UMTS, 12) =
coopWB.Worksheets("Data").Range("B7")

'cellid
Dim CheckUMTScellid As String
CheckUMTScellid = UMTSwb.Worksheets(UMTSwbSheetName(1)).Cells(ODDrow_UMTS,
Val(topicUCol(4))).Value
cellidU(allnum) = CheckUMTScellid

If CheckUMTScellid >= 65535 Then
    coopWB.Worksheets("process").Cells(4+ countUError, 13) = "Cell ID = " &
CheckUMTScellid & " : Range 0-65535 only."
    BSC6900wb.Worksheets(BSC6900ws2).Cells(ODD_UMTS, 13).Value = ""
    countUError = countUError + 1
Else
    BSC6900wb.Worksheets(BSC6900ws2).Cells(ODD_UMTS, 13) =
UMTSwb.Worksheets(UMTSwbSheetName(1)).Cells(ODDrow_UMTS, Val(topicUCol(4))).Value
End If

BSC6900wb.Worksheets(BSC6900ws2).Cells(ODD_UMTS, 14) =
UMTSwb.Worksheets(UMTSwbSheetName(1)).Cells(ODDrow_UMTS, Val(topicUCol(3))).Value

'cellname
Dim CheckUMTScellname As String
CheckUMTScellname = BSC6900wb.Worksheets(BSC6900ws2).Cells(ODD_UMTS, 14).Value
cellnameU(allnum) = CheckUMTScellname

If Len(CheckUMTScellname) <> 10 Then
    coopWB.Worksheets("process").Cells(4+ countUError, 13) = "Cell Name = " &
CheckUMTScellname & " : is invalid."
    countUError = countUError + 1
End If

'*CN Operator Group Index
If Mid(alldigit, 9, 1) = 3 Then 'freg = 3X
    BSC6900wb.Worksheets(BSC6900ws2).Cells(ODD_UMTS, 15) = "0"

```

```

ElseIf Mid(alldigit, 9, 1) = 4 Then 'freg = 4X
    BSC6900wb.Worksheets(BSC6900ws2).Cells(ODD_UMTS, 15) = "2"
ElseIf Mid(alldigit, 9, 1) = 5 Then 'freg = 5X
    BSC6900wb.Worksheets(BSC6900ws2).Cells(ODD_UMTS, 15) = "2"
ElseIf Mid(alldigit, 9, 1) = 6 Then 'freg = 6X
    BSC6900wb.Worksheets(BSC6900ws2).Cells(ODD_UMTS, 15) = "2"
End If

```

```

BSC6900wb.Worksheets(BSC6900ws2).Cells(ODD_UMTS, 16) =
UMTSwb.Worksheets(UMTSwbSheetName(1)).Cells(ODDrow_UMTS, Val(topicUCol(6))).Value
BSC6900wb.Worksheets(BSC6900ws2).Cells(ODD_UMTS, 17) = "TRUE"
BSC6900wb.Worksheets(BSC6900ws2).Cells(ODD_UMTS, 18) =
UMTSwb.Worksheets(UMTSwbSheetName(1)).Cells(ODDrow_UMTS, Val(topicUCol(7))).Value
BSC6900wb.Worksheets(BSC6900ws2).Cells(ODD_UMTS, 19) =
UMTSwb.Worksheets(UMTSwbSheetName(1)).Cells(ODDrow_UMTS, Val(topicUCol(8))).Value
BSC6900wb.Worksheets(BSC6900ws2).Cells(ODD_UMTS, 20) =
UMTSwb.Worksheets(UMTSwbSheetName(1)).Cells(ODDrow_UMTS, Val(topicUCol(9))).Value

```

```

Dim sectorS As String
sectorS = CStr(Left(BSC6900wb.Worksheets(BSC6900ws2).Cells(ODD_UMTS, 14), 5)) & "-"
& CStr(Right(BSC6900wb.Worksheets(BSC6900ws2).Cells(ODD_UMTS, 14), 1))
'MsgBox "sector" & sectorS
Dim PSC As String
PSC = BSC6900wb.Worksheets(BSC6900ws2).Cells(ODD_UMTS, 20).Value
Dim numM As Integer
numM = 0
Dim textS As String
textS = "NO"
Dim message As String
message = "-"
'MsgBox numberS

Do While numM < 10
    'MsgBox "groupS" & groupS(numM)
    If sectorS = groupS(numM) Then
        textS = "YES"
        message = CStr(numM)
    End If
    numM = numM + 1
Loop

'MsgBox sectorS & textS

If textS = "NO" Then
    groupS(numberS) = sectorS

```

```
'
    MsgBox sectorS & textS
'
    MsgBox sectorS
'
    MsgBox numberS
    message = CStr(numberS)
    numberS = numberS + 1
End If
```

```
'MsgBox message
```

```
If message = "0" Then
    PSC0(numberP0) = PSC
    'MsgBox "PSC0" & numberP0 & " : " & PSC0(numberP0)
    numberP0 = numberP0 + 1
ElseIf message = "1" Then
    PSC1(numberP1) = PSC
    numberP1 = numberP1 + 1
ElseIf message = "2" Then
    PSC2(numberP2) = PSC
    numberP2 = numberP2 + 1
ElseIf message = "3" Then
    PSC3(numberP3) = PSC
    numberP3 = numberP3 + 1
ElseIf message = "4" Then
    PSC4(numberP4) = PSC
    numberP4 = numberP4 + 1
ElseIf message = "5" Then
    PSC5(numberP5) = PSC
    numberP5 = numberP5 + 1
End If
```

```
'----- LAC -----'
```

```
Dim rnc As String
```

```
rnc = BSC6900wb.Worksheets(BSC6900ws1).Cells(3, 10)
```

```
If rnc = "3RNCBPL1H" Then
```

```
    If BSC6900wb.Worksheets(BSC6900ws2).Cells(ODD_UMTS, 15) = "0" Then
```

```
        BSC6900wb.Worksheets(BSC6900ws2).Cells(ODD_UMTS, 21) = "5025"
```

```
    ElseIf BSC6900wb.Worksheets(BSC6900ws2).Cells(ODD_UMTS, 15) = "2" Then
```

```
        BSC6900wb.Worksheets(BSC6900ws2).Cells(ODD_UMTS, 21) = "5021"
```

```
    End If
```

```
End If
```

```
If rnc = "3RNCBPL2H" Then
```

```
    If BSC6900wb.Worksheets(BSC6900ws2).Cells(ODD_UMTS, 15) = "0" Then
```

```
BSC6900wb.Worksheets(BSC6900ws2).Cells(ODD_UMTS, 21) = "6025"  
ElseIf BSC6900wb.Worksheets(BSC6900ws2).Cells(ODD_UMTS, 15) = "2" Then  
    BSC6900wb.Worksheets(BSC6900ws2).Cells(ODD_UMTS, 21) = "6021"  
End If  
End If
```

```
If rnc = "3RNCBPL3H" Then  
    If BSC6900wb.Worksheets(BSC6900ws2).Cells(ODD_UMTS, 15) = "0" Then  
        BSC6900wb.Worksheets(BSC6900ws2).Cells(ODD_UMTS, 21) = "7025"  
    ElseIf BSC6900wb.Worksheets(BSC6900ws2).Cells(ODD_UMTS, 15) = "2" Then  
        BSC6900wb.Worksheets(BSC6900ws2).Cells(ODD_UMTS, 21) = "7021"  
    End If  
End If
```

```
If rnc = "3RNCBPL4H" Then  
    If BSC6900wb.Worksheets(BSC6900ws2).Cells(ODD_UMTS, 15) = "0" Then  
        BSC6900wb.Worksheets(BSC6900ws2).Cells(ODD_UMTS, 21) = "4015"  
    ElseIf BSC6900wb.Worksheets(BSC6900ws2).Cells(ODD_UMTS, 15) = "2" Then  
        BSC6900wb.Worksheets(BSC6900ws2).Cells(ODD_UMTS, 21) = "4011"  
    End If  
End If
```

```
If rnc = "3RNCBPL5H" Then  
    If BSC6900wb.Worksheets(BSC6900ws2).Cells(ODD_UMTS, 15) = "0" Then  
        BSC6900wb.Worksheets(BSC6900ws2).Cells(ODD_UMTS, 21) = "5015"  
    ElseIf BSC6900wb.Worksheets(BSC6900ws2).Cells(ODD_UMTS, 15) = "2" Then  
        BSC6900wb.Worksheets(BSC6900ws2).Cells(ODD_UMTS, 21) = "5011"  
    End If  
End If
```

```
If rnc = "3RNCBPL6H" Then  
    If BSC6900wb.Worksheets(BSC6900ws2).Cells(ODD_UMTS, 15) = "0" Then  
        BSC6900wb.Worksheets(BSC6900ws2).Cells(ODD_UMTS, 21) = "6015"  
    ElseIf BSC6900wb.Worksheets(BSC6900ws2).Cells(ODD_UMTS, 15) = "2" Then  
        BSC6900wb.Worksheets(BSC6900ws2).Cells(ODD_UMTS, 21) = "6011"  
    End If  
End If
```

```
If rnc = "3RNCBPL7H" Then  
    If BSC6900wb.Worksheets(BSC6900ws2).Cells(ODD_UMTS, 15) = "0" Then  
        BSC6900wb.Worksheets(BSC6900ws2).Cells(ODD_UMTS, 21) = "10015"  
    ElseIf BSC6900wb.Worksheets(BSC6900ws2).Cells(ODD_UMTS, 15) = "2" Then  
        BSC6900wb.Worksheets(BSC6900ws2).Cells(ODD_UMTS, 21) = "10011"  
    End If  
End If
```

```
If rnc = "3RNCBPL8H" Then
  If BSC6900wb.Worksheets(BSC6900ws2).Cells(ODD_UMTS, 15) = "0" Then
    BSC6900wb.Worksheets(BSC6900ws2).Cells(ODD_UMTS, 21) = "12015"
  ElseIf BSC6900wb.Worksheets(BSC6900ws2).Cells(ODD_UMTS, 15) = "2" Then
    BSC6900wb.Worksheets(BSC6900ws2).Cells(ODD_UMTS, 21) = "12011"
  End If
End If
```

```
If rnc = "3RNCBPL9H" Then
  If BSC6900wb.Worksheets(BSC6900ws2).Cells(ODD_UMTS, 15) = "0" Then
    BSC6900wb.Worksheets(BSC6900ws2).Cells(ODD_UMTS, 21) = "13015"
  ElseIf BSC6900wb.Worksheets(BSC6900ws2).Cells(ODD_UMTS, 15) = "2" Then
    BSC6900wb.Worksheets(BSC6900ws2).Cells(ODD_UMTS, 21) = "13011"
  End If
End If
```

```
If rnc = "3RNCSUK1H" Then
  If BSC6900wb.Worksheets(BSC6900ws2).Cells(ODD_UMTS, 15) = "0" Then
    BSC6900wb.Worksheets(BSC6900ws2).Cells(ODD_UMTS, 21) = "1025"
  ElseIf BSC6900wb.Worksheets(BSC6900ws2).Cells(ODD_UMTS, 15) = "2" Then
    BSC6900wb.Worksheets(BSC6900ws2).Cells(ODD_UMTS, 21) = "1021"
  End If
End If
```

```
If rnc = "3RNCSUK2H" Then
  If BSC6900wb.Worksheets(BSC6900ws2).Cells(ODD_UMTS, 15) = "0" Then
    BSC6900wb.Worksheets(BSC6900ws2).Cells(ODD_UMTS, 21) = "2025"
  ElseIf BSC6900wb.Worksheets(BSC6900ws2).Cells(ODD_UMTS, 15) = "2" Then
    BSC6900wb.Worksheets(BSC6900ws2).Cells(ODD_UMTS, 21) = "2021"
  End If
End If
```

```
If rnc = "3RNCSUK3H" Then
  If BSC6900wb.Worksheets(BSC6900ws2).Cells(ODD_UMTS, 15) = "0" Then
    BSC6900wb.Worksheets(BSC6900ws2).Cells(ODD_UMTS, 21) = "3025"
  ElseIf BSC6900wb.Worksheets(BSC6900ws2).Cells(ODD_UMTS, 15) = "2" Then
    BSC6900wb.Worksheets(BSC6900ws2).Cells(ODD_UMTS, 21) = "3021"
  End If
End If
```

```
If rnc = "3RNCCWD1H" Then
  If BSC6900wb.Worksheets(BSC6900ws2).Cells(ODD_UMTS, 15) = "0" Then
    BSC6900wb.Worksheets(BSC6900ws2).Cells(ODD_UMTS, 21) = "14025"
  ElseIf BSC6900wb.Worksheets(BSC6900ws2).Cells(ODD_UMTS, 15) = "2" Then
    BSC6900wb.Worksheets(BSC6900ws2).Cells(ODD_UMTS, 21) = "14021"
  End If
End If
```

End If
End If

If rnc = "3RNCCWD2H" Then
 If BSC6900wb.Worksheets(BSC6900ws2).Cells(ODD_UMTS, 15) = "0" Then
 BSC6900wb.Worksheets(BSC6900ws2).Cells(ODD_UMTS, 21) = "15025"
 ElseIf BSC6900wb.Worksheets(BSC6900ws2).Cells(ODD_UMTS, 15) = "2" Then
 BSC6900wb.Worksheets(BSC6900ws2).Cells(ODD_UMTS, 21) = "15021"
 End If
End If

If rnc = "3RNCCWD3H" Then
 If BSC6900wb.Worksheets(BSC6900ws2).Cells(ODD_UMTS, 15) = "0" Then
 BSC6900wb.Worksheets(BSC6900ws2).Cells(ODD_UMTS, 21) = "16025"
 ElseIf BSC6900wb.Worksheets(BSC6900ws2).Cells(ODD_UMTS, 15) = "2" Then
 BSC6900wb.Worksheets(BSC6900ws2).Cells(ODD_UMTS, 21) = "16021"
 End If
End If

If rnc = "3RNCTWA1H" Then
 If BSC6900wb.Worksheets(BSC6900ws2).Cells(ODD_UMTS, 15) = "0" Then
 BSC6900wb.Worksheets(BSC6900ws2).Cells(ODD_UMTS, 21) = "18025"
 ElseIf BSC6900wb.Worksheets(BSC6900ws2).Cells(ODD_UMTS, 15) = "2" Then
 BSC6900wb.Worksheets(BSC6900ws2).Cells(ODD_UMTS, 21) = "18021"
 End If
End If

If rnc = "3RNCTWA2H" Then
 If BSC6900wb.Worksheets(BSC6900ws2).Cells(ODD_UMTS, 15) = "0" Then
 BSC6900wb.Worksheets(BSC6900ws2).Cells(ODD_UMTS, 21) = "19025"
 ElseIf BSC6900wb.Worksheets(BSC6900ws2).Cells(ODD_UMTS, 15) = "2" Then
 BSC6900wb.Worksheets(BSC6900ws2).Cells(ODD_UMTS, 21) = "19021"
 End If
End If

If rnc = "3RNCTWA3H" Then
 If BSC6900wb.Worksheets(BSC6900ws2).Cells(ODD_UMTS, 15) = "0" Then
 BSC6900wb.Worksheets(BSC6900ws2).Cells(ODD_UMTS, 21) = "20025"
 ElseIf BSC6900wb.Worksheets(BSC6900ws2).Cells(ODD_UMTS, 15) = "2" Then
 BSC6900wb.Worksheets(BSC6900ws2).Cells(ODD_UMTS, 21) = "20021"
 End If
End If

If rnc = "3RNCTWA4H" Then
 If BSC6900wb.Worksheets(BSC6900ws2).Cells(ODD_UMTS, 15) = "0" Then

```
BSC6900wb.Worksheets(BSC6900ws2).Cells(ODD_UMTS, 21) = "7015"  
Elseif BSC6900wb.Worksheets(BSC6900ws2).Cells(ODD_UMTS, 15) = "2" Then  
    BSC6900wb.Worksheets(BSC6900ws2).Cells(ODD_UMTS, 21) = "7011"  
End If  
End If
```

```
If rnc = "3RNCTWA5H" Then  
    If BSC6900wb.Worksheets(BSC6900ws2).Cells(ODD_UMTS, 15) = "0" Then  
        BSC6900wb.Worksheets(BSC6900ws2).Cells(ODD_UMTS, 21) = "8015"  
    Elseif BSC6900wb.Worksheets(BSC6900ws2).Cells(ODD_UMTS, 15) = "2" Then  
        BSC6900wb.Worksheets(BSC6900ws2).Cells(ODD_UMTS, 21) = "8011"  
    End If  
End If
```

```
If rnc = "3RNCTWA6H" Then  
    If BSC6900wb.Worksheets(BSC6900ws2).Cells(ODD_UMTS, 15) = "0" Then  
        BSC6900wb.Worksheets(BSC6900ws2).Cells(ODD_UMTS, 21) = "9015"  
    Elseif BSC6900wb.Worksheets(BSC6900ws2).Cells(ODD_UMTS, 15) = "2" Then  
        BSC6900wb.Worksheets(BSC6900ws2).Cells(ODD_UMTS, 21) = "9011"  
    End If  
End If
```

```
If rnc = "3RNCTWA7H" Then  
    If BSC6900wb.Worksheets(BSC6900ws2).Cells(ODD_UMTS, 15) = "0" Then  
        BSC6900wb.Worksheets(BSC6900ws2).Cells(ODD_UMTS, 21) = "1015"  
    Elseif BSC6900wb.Worksheets(BSC6900ws2).Cells(ODD_UMTS, 15) = "2" Then  
        BSC6900wb.Worksheets(BSC6900ws2).Cells(ODD_UMTS, 21) = "35011"  
    End If  
End If
```

```
If rnc = "3RNCTWA8H" Then  
    If BSC6900wb.Worksheets(BSC6900ws2).Cells(ODD_UMTS, 15) = "0" Then  
        BSC6900wb.Worksheets(BSC6900ws2).Cells(ODD_UMTS, 21) = "2015"  
    Elseif BSC6900wb.Worksheets(BSC6900ws2).Cells(ODD_UMTS, 15) = "2" Then  
        BSC6900wb.Worksheets(BSC6900ws2).Cells(ODD_UMTS, 21) = "2011"  
    End If  
End If
```

```
If rnc = "3RNCTWA8H" Then  
    If BSC6900wb.Worksheets(BSC6900ws2).Cells(ODD_UMTS, 15) = "0" Then  
        BSC6900wb.Worksheets(BSC6900ws2).Cells(ODD_UMTS, 21) = "3015"  
    Elseif BSC6900wb.Worksheets(BSC6900ws2).Cells(ODD_UMTS, 15) = "2" Then  
        BSC6900wb.Worksheets(BSC6900ws2).Cells(ODD_UMTS, 21) = "3011"  
    End If  
End If
```

```

'-----'

BSC6900wb.Worksheets(BSC6900ws2).Cells(ODD_UMTS, 22) =
UMTSwb.Worksheets(UMTSwbSheetName(1)).Cells(ODDrow_UMTS, Val(topicUCol(11))).Value
BSC6900wb.Worksheets(BSC6900ws2).Cells(ODD_UMTS, 23) = "REQUIRE"
BSC6900wb.Worksheets(BSC6900ws2).Cells(ODD_UMTS, 24) = "2"
BSC6900wb.Worksheets(BSC6900ws2).Cells(ODD_UMTS, 25) =
UMTSwb.Worksheets(UMTSwbSheetName(1)).Cells(ODDrow_UMTS, Val(topicUCol(13))).Value
BSC6900wb.Worksheets(BSC6900ws2).Cells(ODD_UMTS, 26) =
UMTSwb.Worksheets(UMTSwbSheetName(1)).Cells(ODDrow_UMTS, Val(topicUCol(14))).Value
BSC6900wb.Worksheets(BSC6900ws2).Cells(ODD_UMTS, 27) =
UMTSwb.Worksheets(UMTSwbSheetName(1)).Cells(ODDrow_UMTS, Val(topicUCol(15))).Value

```

```

'chip
Dim CheckUMTSchip As String
Dim TMchip(0 To 9) As Variant
Dim TM As Integer
Dim quote As String

```

```

CheckUMTSchip = UMTSwb.Worksheets(UMTSwbSheetName(1)).Cells(ODDrow_UMTS,
Val(topicUCol(16))).Value

```

```

TMchip(0) = "CHIP0"
TMchip(1) = "CHIP256"
TMchip(2) = "CHIP512"
TMchip(3) = "CHIP768"
TMchip(4) = "CHIP1024"
TMchip(5) = "CHIP1280"
TMchip(6) = "CHIP1536"
TMchip(7) = "CHIP1792"
TMchip(8) = "CHIP2048"
TMchip(9) = "CHIP2304"

```

```

quote = "F"
TM = 0
Do While TM < 10
    If CheckUMTSchip = TMchip(TM) Then
        quote = "T"
    End If
    TM = TM + 1
Loop

```

```

If quote = "F" Then
    coopWB.Worksheets("process").Cells(4+countUError, 13) = "Time Offset[chip] = " &
CheckUMTSchip & " : is invalid."

```

```

BSC6900wb.Worksheets(BSC6900ws2).Cells(ODD_UMTS, 28).Value = ""
countUError = countUError + 1
Else
    BSC6900wb.Worksheets(BSC6900ws2).Cells(ODD_UMTS, 28) =
UMTSwb.Worksheets(UMTSwbSheetName(1)).Cells(ODDrow_UMTS, Val(topicUCol(16))).Value
End If

BSC6900wb.Worksheets(BSC6900ws2).Cells(ODD_UMTS, 29) = "430"
BSC6900wb.Worksheets(BSC6900ws2).Cells(ODD_UMTS, 30) = "330"
BSC6900wb.Worksheets(BSC6900ws2).Cells(ODD_UMTS, 31) =
UMTSwb.Worksheets(UMTSwbSheetName(1)).Cells(ODDrow_UMTS, Val(topicUCol(17))).Value

'-----column 32-----'
'AIS or TOT
If Mid(alldigit, 9, 1) = 3 Then
    BSC6900wb.Worksheets(BSC6900ws2).Cells(ODD_UMTS, 32) = "AIS 3G2100 Cell
Template"
Else 'digit = 3
    BSC6900wb.Worksheets(BSC6900ws2).Cells(ODD_UMTS, 32) = "TOT 3G2100 Cell
Template_Unbarred"
End If

BSC6900wb.Worksheets(BSC6900ws2).Cells(ODD_UMTS, 33) = "DEACTIVATED"
BSC6900wb.Worksheets(BSC6900ws2).Cells(ODD_UMTS, 34) =
UMTSwb.Worksheets(UMTSwbSheetName(1)).Cells(ODDrow_UMTS, Val(topicUCol(17))).Value

'sector equipment
SEctor = UMTSwb.Worksheets(UMTSwbSheetName(1)).Cells(ODDrow_UMTS,
Val(topicUCol(17))).Value

If SEctor < 40 Then
    BSC6900wb.Worksheets(BSC6900ws2).Cells(ODD_UMTS, 35) = (CStr(Right(SEctor, 1)) - 1
+ 20)
ElseIf SEctor < 50 Then
    BSC6900wb.Worksheets(BSC6900ws2).Cells(ODD_UMTS, 35) = (CStr(Right(SEctor, 1)) - 1
+ 30)
ElseIf SEctor < 60 Then
    BSC6900wb.Worksheets(BSC6900ws2).Cells(ODD_UMTS, 35) = (CStr(Right(SEctor, 1)) - 1
+ 20)
Else
    BSC6900wb.Worksheets(BSC6900ws2).Cells(ODD_UMTS, 35) = (CStr(Right(SEctor, 1)) - 1
+ 30)
End If

If ODD_UMTS = ODD_UMTS + 1 Then

```

```

        With BSC6900wb.Worksheets(BSC6900ws2)
            .Rows(ODD_UMTS & ":" & .Rows.Count).Delete
        End With
    End If

    allnum = allnum + 1
    ODD_UMTS = ODD_UMTS + 1
    End If
    ODDrow_UMTS = ODDrow_UMTS + 1
    Loop

'----- Ulgroup -----'

Dim Ulgroup(0 To 5) As String
Ulgroup(0) = "0"
Ulgroup(1) = "1"
Ulgroup(2) = "2"
Ulgroup(3) = "3"
Ulgroup(4) = "4"
Ulgroup(5) = "5"

Dim w0, w1, w2, t, tt, rr As Integer
w0 = 0
w1 = 0
w2 = 0
t = 0
tt = 0
rr = 3

Do While rr < ODD_UMTS
    Dim sectorU As Integer
    sectorU = BSC6900wb.Worksheets(BSC6900ws2).Cells(rr, 2).Value
    If w2 < 6 Then
        If sectorU < 40 Then
            'MsgBox "sectorU < 40 : " & sectorU
            BSC6900wb.Worksheets(BSC6900ws2).Cells(rr, 4) = Ulgroup(t)
            w2 = w2 + 1
            If w2 = 6 Then
                t = t + 1
                w2 = 0
            End If
        End If
    End If
    rr = rr + 1
Loop

```

```
'MsgBox t
```

```
rr = 3
```

```
Do While rr < ODD_UMTS
```

```
sectorU= BSC6900wb.Worksheets(BSC6900ws2).Cells(rr, 2).Value
```

```
  If w2 < 6 Then
```

```
    If 40 < sectorU And sectorU < 50 Then
```

```
      'MsgBox "sectorU < 50 : " & sectorU
```

```
      BSC6900wb.Worksheets(BSC6900ws2).Cells(rr, 4) = ULgroup(t)
```

```
      w2 = w2 + 1
```

```
      If w2 = 6 Then
```

```
        t = t + 1
```

```
        w2 = 0
```

```
      End If
```

```
    End If
```

```
  End If
```

```
rr = rr + 1
```

```
Loop
```

```
'MsgBox t
```

```
rr = 3
```

```
Do While rr < ODD_UMTS
```

```
sectorU= BSC6900wb.Worksheets(BSC6900ws2).Cells(rr, 2).Value
```

```
  If w2 < 6 Then
```

```
    If 50 < sectorU And sectorU < 60 Then
```

```
      'MsgBox "sectorU < 60 : " & sectorU
```

```
      BSC6900wb.Worksheets(BSC6900ws2).Cells(rr, 4) = ULgroup(t)
```

```
      w2 = w2 + 1
```

```
      If w2 = 6 Then
```

```
        t = t + 1
```

```
        w2 = 0
```

```
      End If
```

```
    End If
```

```
  End If
```

```
rr = rr + 1
```

```
Loop
```

```
'MsgBox t
```

```
rr = 3
```

```
Do While rr < ODD_UMTS
```

```
sectorU= BSC6900wb.Worksheets(BSC6900ws2).Cells(rr, 2).Value
```

```
  If w2 < 6 Then
```

```

If sectorU > 60 Then
'MsgBox "sectorU > 60 : " & sectorU
    BSC6900wb.Worksheets(BSC6900ws2).Cells(rr, 4) = ULgroup(t)
    w2 = w2 + 1
    If w2 = 6 Then
        t = t + 1
        w2 = 0
    End If
End If
End If
rr = rr + 1
Loop

'-----localcell checking-----'

Dim ij1, ij2, ij3, ij4 As Integer
Dim ij As String
Dim localUdup As Variant
localUdup = Array("", "", "", "", "", "", "", "", "", "", "")

For ij1 = 0 To UBound(localUMTS)
    For ij2 = ij1 + 1 To UBound(localUMTS)
        ij = "yes"
        'MsgBox localUMTS(ij1)
        If localUMTS(ij1) = localUMTS(ij2) And localUMTS(ij1) <> "" And localUMTS(ij2) <> "" Then
            For ij3 = 0 To 9
                If localUdup(ij3) = localUMTS(ij1) Then
                    ij = "no"
                End If
            Next ij3

            If ij = "yes" Then
                localUdup(ij4) = localUMTS(ij1)
                coopWB.Worksheets("process").Cells(4 + countUError, 13) = "Local cell = " &
localUdup(ij4) & " are repeated."
                countUError = countUError + 1
                ij4 = ij4 + 1
            End If
        End If
    Next ij2
Next ij1

'-----Cell ID checking-----'

Dim cd1, cd2, cd3, cd4 As Integer
Dim cdsay As String

```

```

Dim cellid_box As Variant
cellid_box = Array("", "", "", "", "", "", "", "", "", "")

For cd1 = 0 To UBound(cellidU)
    For cd2 = cd1 + 1 To UBound(cellidU)
        MsgBox cellidU(cd1) & "-" & cellidU(cd2)
        cdsay = "yes"
        If cellidU(cd1) = cellidU(cd2) And cellidU(cd1) <> "" And cellidU(cd2) <> "" Then
            For cd3 = 0 To 9
                If cellid_box(cd3) = cellidU(cd1) Then
                    cdsay = "no"
                End If
            Next cd3

            If cdsay = "yes" Then
                cellid_box(cd4) = cellidU(cd1)
                cd4 = cd4 + 1
            End If
        End If
    Next cd2
Next cd1

Dim findCDnum, findCD As Integer
For findCDnum = 0 To UBound(cellid_box)
    For findCD = 0 To UBound(cellidU)
        If cellid_box(findCDnum) = cellidU(findCD) And cellid_box(findCDnum) <> "" And
cellidU(findCD) <> "" Then
            coopWB.Worksheets("process").Cells(4 + countUError, 13) = "Cell ID = " &
cellid_box(findCDnum) _
            & " are repeated values @ cellname : " & cellnameU(findCD)
            countUError = countUError + 1
        End If
    Next findCD
Next findCDnum

```

'----- CellName checking -----'

```

Dim cn1, cn2, cn3, cn4 As Integer
Dim cnsay As String
Dim cellname_box As Variant
cellname_box = Array("", "", "", "", "", "", "", "", "", "")

For cn1 = 0 To UBound(cellnameU)
    For cn2 = cn1 + 1 To UBound(cellnameU)
        cnsay = "Yes"
    
```

```

    If cellnameU(cn1) = cellnameU(cn2) And cellnameU(cn1) <> "" And cellnameU(cn2) <> ""
Then
    For cn3 = 0 To 9
        If cellname_box(cn3) = cellnameU(cn1) Then
            cnsay = "No"
        End If
    Next cn3

    If cnsay = "Yes" Then
        cellname_box(cn4) = cellnameU(cn1)
        coopWB.Worksheets("process").Cells(4+countUError, 13) = "Cell Name = " &
cellname_box(cn4) & " are repeated."
        countUError = countUError + 1
        cn4 = cn4 + 1
    End If
End If
Next cn2
Next cn1

```

'-----DL PSC checking (sector)-----'

```

renewPSC (PSC0)
'MsgBox "0 : " & groupS(0) & PSCdup
If PSCdup > 1 Then
    coopWB.Worksheets("process").Cells(4+countUError, 13) = "PSC of sector = " & groupS(0) &
" are " & PSCdup & " different values."
    countUError = countUError + 1
End If

```

```

renewPSC (PSC1)
'MsgBox "1 : " & groupS(1) & PSCdup
If PSCdup > 1 Then
    coopWB.Worksheets("process").Cells(4+countUError, 13) = "PSC of sector = " & groupS(1) &
" are " & PSCdup & " different values."
    countUError = countUError + 1
End If

```

```

renewPSC (PSC2)
'MsgBox "2 : " & groupS(2) & PSCdup
If PSCdup > 1 Then
    coopWB.Worksheets("process").Cells(4+countUError, 13) = "PSC of sector = " & groupS(2) &
" are " & PSCdup & " different values."
    countUError = countUError + 1
End If

```

```

renewPSC (PSC3)
'MsgBox "3 : " & groupS(3) & PSCdup
If PSCdup > 1 Then
    coopWB.Worksheets("process").Cells(4+countUError, 13) = "PSC of sector = " & groupS(3) &
" are " & PSCdup & " different values."
    countUError = countUError + 1
End If

```

```

renewPSC (PSC4)
'MsgBox "4 : " & groupS(4) & PSCdup
If PSCdup > 1 Then
    coopWB.Worksheets("process").Cells(4+countUError, 13) = "PSC of sector = " & groupS(4) &
" are " & PSCdup & " different values."
    countUError = countUError + 1
End If

```

```

renewPSC (PSC5)
'MsgBox "5 : " & groupS(5) & PSCdup
If PSCdup > 1 Then
    coopWB.Worksheets("process").Cells(4+countUError, 13) = "PSC of sector = " & groupS(5) &
" are " & PSCdup & " different values."
    countUError = countUError + 1
End If

```

'-----DL PSC checking (sitecode)-----'

```

Dim PSC_box(0To5) As Variant
bx1 = 0
renewPSC (PSC0)
If PSCbox(0) <> "-" And PSC_box(bx1) = "" Then
    PSC_box(0) = PSCbox(0)
    'MsgBox "PSC_box" & "(" & bx1 & ")" & PSC_box(bx1)
    bx1 = bx1 + 1
End If

```

```

renewPSC (PSC1)
If PSCbox(0) <> "-" And PSC_box(bx1) = "" Then
    PSC_box(1) = PSCbox(0)
    'MsgBox "PSC_box" & "(" & bx1 & ")" & PSC_box(bx1)
    bx1 = bx1 + 1
End If

```

```

renewPSC (PSC2)
If PSCbox(0) <> "-" And PSC_box(bx1) = "" Then
    PSC_box(2) = PSCbox(0)

```

```

'MsgBox "PSC_box" & "(" & bx1 & ")" & PSC_box(bx1)
bx1= bx1 + 1
End If
renewPSC (PSC3)
If PSCbox(0) <>"-" And PSC_box(bx1)="" Then
PSC_box(3)=PSCbox(0)
'MsgBox "PSC_box" & "(" & bx1 & ")" & PSC_box(bx1)
bx1= bx1 + 1
End If

renewPSC (PSC4)
If PSCbox(0) <>"-" And PSC_box(bx1)="" Then
PSC_box(4)=PSCbox(0)
'MsgBox "PSC_box" & "(" & bx1 & ")" & PSC_box(bx1)
bx1= bx1 + 1
End If

renewPSC (PSC5)
If PSCbox(0) <>"-" And PSC_box(bx1)="" Then
PSC_box(5)=PSCbox(0)
'MsgBox "PSC_box" & "(" & bx1 & ")" & PSC_box(bx1)
bx1= bx1 + 1
End If

Dim repeatedPSC As Variant
repeatedPSC=PSC_box
repeatedPSC=PSCrepeat(PSC_box)

Dim Errorstr As String
Errorstr = "Duplicate PSC between sector : "
Dim Numi As Integer

For Numi = 0 To UBound(repeatedPSC)
If repeatedPSC(Numi) <>"" Then
coopWB.Worksheets("process").Cells(4+ countUError, 13) = Errorstr & groupS(Numi) & "=
" & repeatedPSC(Numi)
countUError = countUError + 1
End If
Next Numi

'set gridlines
BSC6900wb.Worksheets(BSC6900ws2).Activate
ActiveSheet.UsedRange.Select
Selection.Borders(xlDiagonalDown).LineStyle = xlNone
Selection.Borders(xlDiagonalUp).LineStyle = xlNone

```

```

With Selection.Borders(xlEdgeLeft)
    .LineStyle = xlContinuous
    .ColorIndex = 0
    .TintAndShade = 0
    .Weight = xlThin
End With
With Selection.Borders(xlEdgeTop)
    .LineStyle = xlContinuous
    .ColorIndex = 0
    .TintAndShade = 0
    .Weight = xlThin
End With
With Selection.Borders(xlEdgeBottom)
    .LineStyle = xlContinuous
    .ColorIndex = 0
    .TintAndShade = 0
    .Weight = xlThin
End With
With Selection.Borders(xlEdgeRight)
    .LineStyle = xlContinuous
    .ColorIndex = 0
    .TintAndShade = 0
    .Weight = xlThin
End With
With Selection.Borders(xlInsideVertical)
    .LineStyle = xlContinuous
    .ColorIndex = 0
    .TintAndShade = 0
    .Weight = xlThin
End With
With Selection.Borders(xlInsideHorizontal)
    .LineStyle = xlContinuous
    .ColorIndex = 0
    .TintAndShade = 0
    .Weight = xlThin
End With

```

----- BSC6900 LTE1800 -----

```

Dim ur, ur2 As Integer
ur2 = 0
ur = 0

```

```

Dim allgroup(0 To 4) As String
allgroup(0) = "13"

```

```

allgroup(1) = "10"
allgroup(2) = "11"
allgroup(3) = "14"
allgroup(4) = "15"

```

```
BSC6900ws3 = "LTE Cell"
```

```

BSC6900wb.Worksheets(BSC6900ws3).Rows("3:" & Rows.Count).ClearContents
BSC6900wb.Worksheets(BSC6900ws3).Rows("3:" & Rows.Count).Interior.Color = xlNone
BSC6900wb.Worksheets(BSC6900ws3).Cells.HorizontalAlignment = xlCenter

```

```
'set font and size
```

```

With BSC6900wb.Worksheets(BSC6900ws3)
    .Cells.Font.Name = "Arial"
    .Cells.Font.Size = 10
End With

```

```
Count_ODD_LTE_18 = 0
```

```
ODDrow_LTE18 = 5
```

```
ODD_L18 = 3
```

```
L18num = 0
```

```
If topicLCol(2) <> "" Then
```

```

    Do While LTEwb.Worksheets(LTEwbSheetName(1)).Cells(ODDrow_LTE18,
Val(topicLCol(2))) <> ""

```

```

        If LTEwb.Worksheets(LTEwbSheetName(1)).Cells(ODDrow_LTE18,
Val(topicLCol(2))).Value = inputsitecode Then

```

```
            'MsgBox ODDrow_LTE18
```

```
            L18num = L18num + 1
```

```

            BSC6900wb.Worksheets(BSC6900ws3).Cells(ODD_L18, 1) =
planwb.Worksheets(planwbSheetName(1)).Cells(siterow, Val(topicCol(1))).Value

```

```
            BSC6900wb.Worksheets(BSC6900ws3).Cells(ODD_L18, 2) =
```

```

            LTEwb.Worksheets(LTEwbSheetName(1)).Cells(ODDrow_LTE18, Val(topicLCol(5))).Value

```

```
            'localcell
```

```
            Dim CheckLTElocalcell As String
```

```
            BSC6900wb.RefreshAll
```

```
            CheckLTElocalcell = BSC6900wb.Worksheets(BSC6900ws3).Cells(ODD_L18, 2).Value
```

```
            localLTE18(allIN) = CheckLTElocalcell
```

```
            If Len(CheckLTElocalcell) <> 3 Then
```

```

                coopWB.Worksheets("process").Cells(4+countLError, 14) = "Local Cell ID = " &
CheckLTElocalcell & " : is invalid."

```

```

        countLError = countLError + 1
    End If

    BSC6900wb.Worksheets(BSC6900ws3).Cells(ODD_L18, 3) =
LTEwb.Worksheets(LTEwbSheetName(1)).Cells(ODDrow_LTE18, Val(topicLCol(4))).Value

    'cellname
    Dim CheckLTEcellname As String
    CheckLTEcellname = BSC6900wb.Worksheets(BSC6900ws3).Cells(ODD_L18, 3).Value
    cellLTE18(allIN) = CheckLTEcellname

    If Len(CheckLTEcellname) <> 10 Then
        coopWB.Worksheets("process").Cells(4+countLError, 14) = "Cell Name = " &
CheckLTEcellname & " : is invalid."
        countLError = countLError + 1
    End If

    'cellid
    Dim CheckLTEcellid As String
    CheckLTEcellid = LTEwb.Worksheets(LTEwbSheetName(1)).Cells(ODDrow_LTE18,
Val(topicLCol(6))).Value
    cellidLTE18(allIN) = CheckLTEcellid

    If CheckLTEcellid >= 255 Then
        coopWB.Worksheets("process").Cells(4+countLError, 14) = "Cell ID = " &
CheckLTEcellid & " : Range 0-255 only."
        BSC6900wb.Worksheets(BSC6900ws3).Cells(ODD_L18, 4).Value = ""
        countLError = countLError + 1
    Else
        BSC6900wb.Worksheets(BSC6900ws3).Cells(ODD_L18, 4) =
LTEwb.Worksheets(LTEwbSheetName(1)).Cells(ODDrow_LTE18, Val(topicLCol(6))).Value
    End If

    BSC6900wb.Worksheets(BSC6900ws3).Cells(ODD_L18, 5) = "0"
    BSC6900wb.Worksheets(BSC6900ws3).Cells(ODD_L18, 6) =
"CELL_NOT_RESERVED_FOR_OP"
    BSC6900wb.Worksheets(BSC6900ws3).Cells(ODD_L18, 7) = "0"
    BSC6900wb.Worksheets(BSC6900ws3).Cells(ODD_L18, 8) =
"CELL_MME_CFG_NUM_4"
    BSC6900wb.Worksheets(BSC6900ws3).Cells(ODD_L18, 9) = "1"
    BSC6900wb.Worksheets(BSC6900ws3).Cells(ODD_L18, 10) = "32768"
    BSC6900wb.Worksheets(BSC6900ws3).Cells(ODD_L18, 11) = "2"
    BSC6900wb.Worksheets(BSC6900ws3).Cells(ODD_L18, 12) = "32768"
    BSC6900wb.Worksheets(BSC6900ws3).Cells(ODD_L18, 13) = "3"
    BSC6900wb.Worksheets(BSC6900ws3).Cells(ODD_L18, 14) = "32768"

```

```

BSC6900wb.Worksheets(BSC6900ws3).Cells(ODD_L18, 15) = "16"
BSC6900wb.Worksheets(BSC6900ws3).Cells(ODD_L18, 16) = "32768"
BSC6900wb.Worksheets(BSC6900ws3).Cells(ODD_L18, 17) =
LTEwb.Worksheets(LTEwbSheetName(1)).Cells(ODDrow_LTE18, Val(topicLCol(8))).Value
BSC6900wb.Worksheets(BSC6900ws3).Cells(ODD_L18, 18) = "BOOLEAN_FALSE"
BSC6900wb.Worksheets(BSC6900ws3).Cells(ODD_L18, 19) = "NORMAL_CP"
BSC6900wb.Worksheets(BSC6900ws3).Cells(ODD_L18, 20) = "NORMAL_CP"
BSC6900wb.Worksheets(BSC6900ws3).Cells(ODD_L18, 21) = "3"
BSC6900wb.Worksheets(BSC6900ws3).Cells(ODD_L18, 22) = "CFG"

```

```

'-----column 23,24-----'
'Freq of UL and DL
allLdigit = LTEwb.Worksheets(LTEwbSheetName(1)).Cells(ODDrow_LTE18,
Val(topicLCol(4))).Value

```

```

If Mid(allLdigit, 7, 2) = 18 Then 'L1800
    BSC6900wb.Worksheets(BSC6900ws3).Cells(ODD_L18, 23) = "19425"
    BSC6900wb.Worksheets(BSC6900ws3).Cells(ODD_L18, 24) = "1425"
ElseIf Mid(allLdigit, 7, 2) = 21 Then 'L2100
    BSC6900wb.Worksheets(BSC6900ws3).Cells(ODD_L18, 23) = "18350"
    BSC6900wb.Worksheets(BSC6900ws3).Cells(ODD_L18, 24) = "350"
ElseIf Mid(allLdigit, 7, 2) = 9 Then 'L900
    BSC6900wb.Worksheets(BSC6900ws3).Cells(ODD_L18, 23) = "21650"
    BSC6900wb.Worksheets(BSC6900ws3).Cells(ODD_L18, 24) = "3650"
End If

```

```

'-----column 25,26-----'
'BandWidth
BW = LTEwb.Worksheets(LTEwbSheetName(1)).Cells(ODDrow_LTE18,
Val(topicLCol(14))).Value

```

```

If BW = "15MHz" Then 'L1800
    BSC6900wb.Worksheets(BSC6900ws3).Cells(ODD_L18, 25) = "CELL_BW_N75"
    BSC6900wb.Worksheets(BSC6900ws3).Cells(ODD_L18, 26) = "CELL_BW_N75"
ElseIf BW = "10MHz" Then 'L2100
    BSC6900wb.Worksheets(BSC6900ws3).Cells(ODD_L18, 25) = "CELL_BW_N50"
    BSC6900wb.Worksheets(BSC6900ws3).Cells(ODD_L18, 26) = "CELL_BW_N50"
ElseIf BW = "5MHz" Then 'L900
    BSC6900wb.Worksheets(BSC6900ws3).Cells(ODD_L18, 25) = "CELL_BW_N25"
    BSC6900wb.Worksheets(BSC6900ws3).Cells(ODD_L18, 26) = "CELL_BW_N25"
End If

```

```

'PCI
Dim CheckPCI As String

```

```

    CheckPCI = LTEwb.Worksheets(LTEwbSheetName(1)).Cells(ODDrow_LTE18,
Val(topicLCol(10))).Value
    PCI_LTE18(allIN) = CheckPCI

    If CheckPCI >= 503 Then
        coopWB.Worksheets("process").Cells(4+countLError, 14) = "PCI = " & CheckPCI & "
: Range 0-503 only."
        BSC6900wb.Worksheets(BSC6900ws3).Cells(ODD_L18, 27).Value = ""
        countLError = countLError + 1
    Else
        BSC6900wb.Worksheets(BSC6900ws3).Cells(ODD_L18, 27) =
LTEwb.Worksheets(LTEwbSheetName(1)).Cells(ODDrow_LTE18, Val(topicLCol(10))).Value
    End If

    BSC6900wb.Worksheets(BSC6900ws3).Cells(ODD_L18, 28) = "1"
    BSC6900wb.Worksheets(BSC6900ws3).Cells(ODD_L18, 29) = "CELL_DEACTIVE"
    BSC6900wb.Worksheets(BSC6900ws3).Cells(ODD_L18, 30) = "CELL_UNBLOCK"
    BSC6900wb.Worksheets(BSC6900ws3).Cells(ODD_L18, 31) = "" 'EMPTY
    BSC6900wb.Worksheets(BSC6900ws3).Cells(ODD_L18, 32) = "CELL_FDD"
    BSC6900wb.Worksheets(BSC6900ws3).Cells(ODD_L18, 33) = "" 'EMPTY
    BSC6900wb.Worksheets(BSC6900ws3).Cells(ODD_L18, 34) = "" 'EMPTY
    BSC6900wb.Worksheets(BSC6900ws3).Cells(ODD_L18, 35) = "dB0"
    BSC6900wb.Worksheets(BSC6900ws3).Cells(ODD_L18, 36) = "dB0"

    'RootSequenceldx
    Dim CheckRSI As String
    CheckRSI = LTEwb.Worksheets(LTEwbSheetName(1)).Cells(ODDrow_LTE18,
Val(topicLCol(11))).Value
    RSI_LTE18(allIN) = CheckRSI

    If CheckRSI >= 837 Then
        coopWB.Worksheets("process").Cells(4+countLError, 14) = "RootSequenceldx = " &
CheckRSI & " : Range 0-837 only."
        BSC6900wb.Worksheets(BSC6900ws3).Cells(ODD_L18, 37).Value = ""
        countLError = countLError + 1
    Else
        BSC6900wb.Worksheets(BSC6900ws3).Cells(ODD_L18, 37) =
LTEwb.Worksheets(LTEwbSheetName(1)).Cells(ODDrow_LTE18, Val(topicLCol(11))).Value
    End If

    BSC6900wb.Worksheets(BSC6900ws3).Cells(ODD_L18, 38) = "LOW_SPEED"
    BSC6900wb.Worksheets(BSC6900ws3).Cells(ODD_L18, 39) = "0"
    BSC6900wb.Worksheets(BSC6900ws3).Cells(ODD_L18, 40) =
LTEwb.Worksheets(LTEwbSheetName(1)).Cells(ODDrow_LTE18, Val(topicLCol(15))).Value
    BSC6900wb.Worksheets(BSC6900ws3).Cells(ODD_L18, 41) = "NOT_CFG"

```

```

BSC6900wb.Worksheets(BSC6900ws3).Cells(ODD_L18, 42) = "" 'EMPTY
BSC6900wb.Worksheets(BSC6900ws3).Cells(ODD_L18, 43) = "" 'EMPTY

'column 44
Dim rngpower As Range
Set rngpower = LTEwb.Worksheets(LTEwbSheetName(1)).Cells(ODDrow_LTE18,
Val(topicLCol(16)))
rngpower = Evaluate(rngpower.Value & "*10")
BSC6900wb.Worksheets(BSC6900ws3).Cells(ODD_L18, 44) = rngpower

BSC6900wb.Worksheets(BSC6900ws3).Cells(ODD_L18, 45) = "6"
BSC6900wb.Worksheets(BSC6900ws3).Cells(ODD_L18, 46) = "NOT_CFG"
BSC6900wb.Worksheets(BSC6900ws3).Cells(ODD_L18, 47) = "" 'EMPTY
BSC6900wb.Worksheets(BSC6900ws3).Cells(ODD_L18, 48) = "NOT_CFG"
BSC6900wb.Worksheets(BSC6900ws3).Cells(ODD_L18, 49) = "" 'EMPTY
BSC6900wb.Worksheets(BSC6900ws3).Cells(ODD_L18, 50) = "BOOLEAN_FALSE"
BSC6900wb.Worksheets(BSC6900ws3).Cells(ODD_L18, 51) = "" 'EMPTY
BSC6900wb.Worksheets(BSC6900ws3).Cells(ODD_L18, 52) = "2T2R"
BSC6900wb.Worksheets(BSC6900ws3).Cells(ODD_L18, 53) = "CRS_PORT_2"
BSC6900wb.Worksheets(BSC6900ws3).Cells(ODD_L18, 54) = "" 'EMPTY
BSC6900wb.Worksheets(BSC6900ws3).Cells(ODD_L18, 55) = "NO_COMPRESSION"
BSC6900wb.Worksheets(BSC6900ws3).Cells(ODD_L18, 56) = "0"

'column 57
'LTE cell template
allLdigit = LTEwb.Worksheets(LTEwbSheetName(1)).Cells(ODDrow_LTE18,
Val(topicLCol(4))).Value
If Mid(allLdigit, 7, 2) = 18 Then 'L1800
    BSC6900wb.Worksheets(BSC6900ws3).Cells(ODD_L18, 57) =
ThisWorkbook.Worksheets("Data").Range("B2").Value
ElseIf Mid(allLdigit, 7, 2) = 21 Then 'L2100
    BSC6900wb.Worksheets(BSC6900ws3).Cells(ODD_L18, 57) =
ThisWorkbook.Worksheets("Data").Range("B3").Value
ElseIf Mid(allLdigit, 8, 2) = 9 Then 'L900
    BSC6900wb.Worksheets(BSC6900ws3).Cells(ODD_L18, 57) =
ThisWorkbook.Worksheets("Data").Range("B4").Value
End If

BSC6900wb.Worksheets(BSC6900ws3).Cells(ODD_L18, 58) =
LTEwb.Worksheets(LTEwbSheetName(1)).Cells(ODDrow_LTE18, Val(topicLCol(5))).Value
BSC6900wb.Worksheets(BSC6900ws3).Cells(ODD_L18, 59) = "BOOLEAN_TRUE"
BSC6900wb.Worksheets(BSC6900ws3).Cells(ODD_L18, 60) = "DEFAULTMODE"
BSC6900wb.Worksheets(BSC6900ws3).Cells(ODD_L18, 61) =
LTEwb.Worksheets(LTEwbSheetName(1)).Cells(ODDrow_LTE18, Val(topicLCol(5))).Value
BSC6900wb.Worksheets(BSC6900ws3).Cells(ODD_L18, 62) = "OFF"

```

```

BSC6900wb.Worksheets(BSC6900ws3).Cells(ODD_L18, 63) = "ms40"
BSC6900wb.Worksheets(BSC6900ws3).Cells(ODD_L18, 64) =
LTEwb.Worksheets(LTEwbSheetName(1)).Cells(ODDrow_LTE18, Val(topicLCol(5))).Value
BSC6900wb.Worksheets(BSC6900ws3).Cells(ODD_L18, 65) = "OFF"
BSC6900wb.Worksheets(BSC6900ws3).Cells(ODD_L18, 66) = "DB_3_P_A"
BSC6900wb.Worksheets(BSC6900ws3).Cells(ODD_L18, 67) =
LTEwb.Worksheets(LTEwbSheetName(1)).Cells(ODDrow_LTE18, Val(topicLCol(5))).Value
BSC6900wb.Worksheets(BSC6900ws3).Cells(ODD_L18, 68) = "MOD3"
BSC6900wb.Worksheets(BSC6900ws3).Cells(ODD_L18, 69) =
LTEwb.Worksheets(LTEwbSheetName(1)).Cells(ODDrow_LTE18, Val(topicLCol(5))).Value

```

```

'column 70
'sector equipment
SECTOR = LTEwb.Worksheets(LTEwbSheetName(1)).Cells(ODDrow_LTE18,
Val(topicLCol(5))).Value

```

```

If Right(SECTOR, 1) = 1 Then
    BSC6900wb.Worksheets(BSC6900ws3).Cells(ODD_L18, 70) = (CStr(Right(SECTOR, 1)) -
1 + 50)
ElseIf Right(SECTOR, 1) = 2 Then
    BSC6900wb.Worksheets(BSC6900ws3).Cells(ODD_L18, 70) = (CStr(Right(SECTOR, 1)) -
1 + 50)
ElseIf Right(SECTOR, 1) = 3 Then
    BSC6900wb.Worksheets(BSC6900ws3).Cells(ODD_L18, 70) = (CStr(Right(SECTOR, 1)) -
1 + 50)
ElseIf Right(SECTOR, 1) = 4 Then
    BSC6900wb.Worksheets(BSC6900ws3).Cells(ODD_L18, 70) = (CStr(Right(SECTOR, 1)) -
1 + 50)
ElseIf Right(SECTOR, 1) = 5 Then
    BSC6900wb.Worksheets(BSC6900ws3).Cells(ODD_L18, 70) = (CStr(Right(SECTOR, 1)) -
1 + 50)
ElseIf Right(SECTOR, 1) = 6 Then
    BSC6900wb.Worksheets(BSC6900ws3).Cells(ODD_L18, 70) = (CStr(Right(SECTOR, 1)) -
1 + 50)
End If

```

```

'column 71
'band group
If ur < 6 Then
    BSC6900wb.Worksheets(BSC6900ws3).Cells(ODD_L18, 71) = allgroup(ur2)
    ur = ur + 1
    If ur = 6 Then
        ur2 = ur2 + 1
        ur = 0
    End If
End If

```

```

        End If

        allN = allN + 1
        ODD_L18 = ODD_L18 + 1
        End If

        ODDrow_LTE18 = ODDrow_LTE18 + 1
        Loop

    End If

'----- localcell checking -----'

    Dim ci1, ci2, ci3, ci4 As Integer
    Dim cii As String
    Dim localdup As Variant
    localdup = Array("", "", "", "", "", "", "", "", "", "")

    For ci1 = 0 To UBound(localLTE18)
        For ci2 = ci1 + 1 To UBound(localLTE18)
            cii = "yes"
            If localLTE18(ci1) = localLTE18(ci2) And localLTE18(ci1) <> "" And localLTE18(ci2) <> ""
Then
                For ci3 = 0 To 9
                    If localdup(ci3) = localLTE18(ci1) Then
                        cii = "no"
                    End If
                Next ci3

                If cii = "yes" Then
                    localdup(ci4) = localLTE18(ci1)
                    coopWB.Worksheets("process").Cells(4+countLError, 14) = "Local cell = " &
localdup(ci4) & " are repeated."
                    localLTE18(ci1) = vbNullString
                    countLError = countLError + 1
                    ci4 = ci4 + 1
                End If
            End If
        Next ci2
    Next ci1

'----- CellName checking -----'

    Dim ni1, ni2, ni3, ni4 As Integer
    Dim nii As String

```

```

Dim cellnamedup As Variant
cellnamedup = Array("", "", "", "", "", "", "", "", "", "", "")

For ni1 = 0 To UBound(cellLTE18)
  For ni2 = ni1 + 1 To UBound(cellLTE18)
    nii = "yes"
    If cellLTE18(ni1) = cellLTE18(ni2) And cellLTE18(ni1) <> "" And cellLTE18(ni2) <> "" Then
      For ni3 = 0 To 9
        If cellnamedup(ni3) = cellLTE18(ni1) Then
          nii = "no"
        End If
      Next ni3

      If nii = "yes" Then
        cellnamedup(ni4) = cellLTE18(ni1)
        'MsgBox cellnamedup(ni4)
        coopWB.Worksheets("process").Cells(4+countLError, 14) = "CellName = " &
cellnamedup(ni4) & " are repeated."
        countLError = countLError + 1
        ni4 = ni4 + 1
      End If
    End If
  Next ni2
Next ni1

'----- CellID checking -----'

Dim ai1, ai2, ai3, ai4 As Integer
Dim aii As String
Dim celliddup As Variant
celliddup = Array("", "", "", "", "", "", "", "", "", "", "")

For ai1 = 0 To UBound(cellidLTE18)
  For ai2 = ai1 + 1 To UBound(cellidLTE18)
    'MsgBox cellidLTE18(ai1) & "-" & cellidLTE18(ai2)
    aii = "yes"
    If cellidLTE18(ai1) = cellidLTE18(ai2) And cellidLTE18(ai1) <> "" And cellidLTE18(ai2) <> ""
Then
      For ai3 = 0 To 9
        If celliddup(ai3) = cellidLTE18(ai1) Then
          aii = "no"
        End If
      Next ai3

      If aii = "yes" Then

```

```

        celliddup(ai4) = cellidLTE18(ai1)
        'MsgBox celliddup(ai4)
        ai4 = ai4 + 1
    End If
End If
Next ai2
Next ai1

Dim findIDdupnum, findIDnum As Integer
For findIDdupnum = 0 To UBound(celliddup)
    For findIDnum = 0 To UBound(cellidLTE18)
        If celliddup(findIDdupnum) = cellidLTE18(findIDnum) And celliddup(findIDdupnum) <> ""
And cellidLTE18(findIDnum) <> "" Then
            'MsgBox cellLTE18(findPnum)
            coopWB.Worksheets("process").Cells(4+countLError, 14) = "Cell ID = " &
celliddup(findIDdupnum)_
            & " are repeated values @ cellname : " & cellLTE18(findIDnum)
            countLError = countLError + 1
        End If
    Next findIDnum
Next findIDdupnum

'----- PCI checking -----'

Dim xi1, xi2, xi3, xi4 As Integer
Dim xii As String
Dim PCIdup As Variant
PCIdup = Array("", "", "", "", "", "", "", "", "", "", "")

For xi1 = 0 To UBound(PCI_LTE18)
    For xi2 = xi1 + 1 To UBound(PCI_LTE18)
        xii = "yes"
        If PCI_LTE18(xi1) = PCI_LTE18(xi2) And PCI_LTE18(xi1) <> "" And PCI_LTE18(xi2) <> ""
Then
            'MsgBox PCI_LTE18(xi1) & "=" & PCI_LTE18(xi2)
            For xi3 = 0 To 9
                If PCIdup(xi3) = PCI_LTE18(xi1) Then
                    xii = "no"
                End If
            Next xi3

            If xii = "yes" Then
                PCIdup(xi4) = PCI_LTE18(xi1)
                'MsgBox PCIdup(xi4)
                xi4 = xi4 + 1
            End If
        End If
    Next xi2
Next xi1

```

```

        End If
    End If
Next xi2
Next xi1

Dim findPdupnum, findPnum As Integer
For findPdupnum = 0 To UBound(PCIdup)
    For findPnum = 0 To UBound(PCI_LTE18)
        If PCIdup(findPdupnum) = PCI_LTE18(findPnum) And PCIdup(findPdupnum) <> "" And
PCI_LTE18(findPnum) <> "" Then
            'MsgBox cellLTE18(findPnum)
            coopWB.Worksheets("process").Cells(4+countLError, 14) = "PCI = " &
PCIIdup(findPdupnum)_
            & " are repeated values @ cellname : " & cellLTE18(findPnum)
            countLError = countLError + 1
        End If
    Next findPnum
Next findPdupnum

'----- Rootsequenceind checking -----'

Dim yi1, yi2, yi3, yi4 As Integer
Dim yii As String
Dim RSIdup As Variant
RSIdup = Array("", "", "", "", "", "", "", "", "", "", "")

For yi1 = 0 To UBound(RSI_LTE18)
    For yi2 = yi1 + 1 To UBound(RSI_LTE18)
        yii = "Yep"
        If RSI_LTE18(yi1) = RSI_LTE18(yi2) And RSI_LTE18(yi1) <> "" And RSI_LTE18(yi2) <> ""
Then
            For yi3 = 0 To 9
                If RSIdup(yi3) = RSI_LTE18(yi1) Then
                    yii = "Nope"
                End If
            Next yi3

            If yii = "Yep" Then
                RSIdup(yi4) = RSI_LTE18(yi1)
                yi4 = yi4 + 1
            End If
        End If
    Next yi2
Next yi1

```

```

Dim findRdupnum, findRnum As Integer
For findRdupnum = 0 To UBound(RSIdup)
    For findRnum = 0 To UBound(RSI_LTE18)
        If RSIdup(findRdupnum) = RSI_LTE18(findRnum) And RSIdup(findRdupnum) <> "" And
RSI_LTE18(findRnum) <> "" Then
            'MsgBox cellLTE18(findRnum)
            coopWB.Worksheets("process").Cells(4+countLError, 14) = "RSI = " &
RSIdup(findRdupnum)_
            & " are repeated values @ cellname : " & cellLTE18(findRnum)
            countLError = countLError + 1
        End If
    Next findRnum
Next findRdupnum

```

'----- BSC6900LTE2100 -----'

```

' MsgBox "L18 num = " & L18num
' MsgBox "Next row is " & ODD_L18
lastrow = ODD_L18

countL18 = 0
L21num = 0

Do While countL18 < L18num
    L21num = L21num + 1
' MsgBox "Next Row is " & lastrow + a1
    BSC6900wb.Worksheets(BSC6900ws3).Cells(lastrow + countL18, 1) =
BSC6900wb.Worksheets(BSC6900ws3).Cells(ODD_L18 - L18num, 1).Value
    BSC6900wb.Worksheets(BSC6900ws3).Cells(lastrow + countL18, 2) =
CStr(BSC6900wb.Worksheets(BSC6900ws3).Cells(ODD_L18 - L18num, 2).Value) + 20

    allcellname = BSC6900wb.Worksheets(BSC6900ws3).Cells(ODD_L18 - L18num, 3).Value
    Rightcellname = Right(allcellname, 2)
    Leftcellname = Left(allcellname, 6)
    cellname = Leftcellname + "21" + Rightcellname

    BSC6900wb.Worksheets(BSC6900ws3).Cells(lastrow + countL18, 3) = cellname
    BSC6900wb.Worksheets(BSC6900ws3).Cells(lastrow + countL18, 4) =
CStr(BSC6900wb.Worksheets(BSC6900ws3).Cells(ODD_L18 - L18num, 2).Value) + 20
    BSC6900wb.Worksheets(BSC6900ws3).Cells(lastrow + countL18, 5) = "0"
    BSC6900wb.Worksheets(BSC6900ws3).Cells(lastrow + countL18, 6) =
"CELL_NOT_RESERVED_FOR_OP"
    BSC6900wb.Worksheets(BSC6900ws3).Cells(lastrow + countL18, 7) = "0"
    BSC6900wb.Worksheets(BSC6900ws3).Cells(lastrow + countL18, 8) =
"CELL_MME_CFG_NUM_4"

```

```

BSC6900wb.Worksheets(BSC6900ws3).Cells(lastrow + countL18, 9) = "1"
BSC6900wb.Worksheets(BSC6900ws3).Cells(lastrow + countL18, 10) = "32768"
BSC6900wb.Worksheets(BSC6900ws3).Cells(lastrow + countL18, 11) = "2"
BSC6900wb.Worksheets(BSC6900ws3).Cells(lastrow + countL18, 12) = "32768"
BSC6900wb.Worksheets(BSC6900ws3).Cells(lastrow + countL18, 13) = "3"
BSC6900wb.Worksheets(BSC6900ws3).Cells(lastrow + countL18, 14) = "32768"
BSC6900wb.Worksheets(BSC6900ws3).Cells(lastrow + countL18, 15) = "16"
BSC6900wb.Worksheets(BSC6900ws3).Cells(lastrow + countL18, 16) = "32768"
BSC6900wb.Worksheets(BSC6900ws3).Cells(lastrow + countL18, 17) =
BSC6900wb.Worksheets(BSC6900ws3).Cells(ODD_L18 - L18num, 17).Value
BSC6900wb.Worksheets(BSC6900ws3).Cells(lastrow + countL18, 18) = "BOOLEAN_FALSE"
BSC6900wb.Worksheets(BSC6900ws3).Cells(lastrow + countL18, 19) = "NORMAL_CP"
BSC6900wb.Worksheets(BSC6900ws3).Cells(lastrow + countL18, 20) = "NORMAL_CP"
BSC6900wb.Worksheets(BSC6900ws3).Cells(lastrow + countL18, 21) = "1"
BSC6900wb.Worksheets(BSC6900ws3).Cells(lastrow + countL18, 22) = "CFG"

```

'-----column 23,24-----'

'Freq of UL and DL

Dim redigit As String

redigit = BSC6900wb.Worksheets(BSC6900ws3).Cells(ODD_L18, 3).Value

If Mid(redigit, 7, 2) = 18 Then 'L1800

BSC6900wb.Worksheets(BSC6900ws3).Cells(lastrow + countL18, 23) = "19425"

BSC6900wb.Worksheets(BSC6900ws3).Cells(lastrow + countL18, 24) = "1425"

Elseif Mid(redigit, 7, 2) = 21 Then 'L2100

BSC6900wb.Worksheets(BSC6900ws3).Cells(lastrow + countL18, 23) = "18350"

BSC6900wb.Worksheets(BSC6900ws3).Cells(lastrow + countL18, 24) = "350"

Elseif Mid(redigit, 7, 2) = 9 Then 'L900

BSC6900wb.Worksheets(BSC6900ws3).Cells(lastrow + countL18, 23) = "21650"

BSC6900wb.Worksheets(BSC6900ws3).Cells(lastrow + countL18, 24) = "3650"

End If

'-----column 25,26-----'

'BandWidth

BSC6900wb.Worksheets(BSC6900ws3).Cells(lastrow + countL18, 25) = "CELL_BW_N50"

BSC6900wb.Worksheets(BSC6900ws3).Cells(lastrow + countL18, 26) = "CELL_BW_N50"

BSC6900wb.Worksheets(BSC6900ws3).Cells(lastrow + countL18, 27) =

BSC6900wb.Worksheets(BSC6900ws3).Cells(ODD_L18 - L18num, 27).Value

BSC6900wb.Worksheets(BSC6900ws3).Cells(lastrow + countL18, 28) = "1"

BSC6900wb.Worksheets(BSC6900ws3).Cells(lastrow + countL18, 29) = "CELL_DEACTIVE"

BSC6900wb.Worksheets(BSC6900ws3).Cells(lastrow + countL18, 30) = "CELL_UNBLOCK"

BSC6900wb.Worksheets(BSC6900ws3).Cells(lastrow + countL18, 31) = "" 'EMPTY

BSC6900wb.Worksheets(BSC6900ws3).Cells(lastrow + countL18, 32) = "CELL_FDD"

BSC6900wb.Worksheets(BSC6900ws3).Cells(lastrow + countL18, 33) = "" 'EMPTY

```

BSC6900wb.Worksheets(BSC6900ws3).Cells(lastrow + countL18, 34) = "" 'EMPTY
BSC6900wb.Worksheets(BSC6900ws3).Cells(lastrow + countL18, 35) = "dB0"
BSC6900wb.Worksheets(BSC6900ws3).Cells(lastrow + countL18, 36) = "dB0"
BSC6900wb.Worksheets(BSC6900ws3).Cells(lastrow + countL18, 37) =
BSC6900wb.Worksheets(BSC6900ws3).Cells(ODD_L18 - L18num, 37).Value
BSC6900wb.Worksheets(BSC6900ws3).Cells(lastrow + countL18, 38) = "LOW_SPEED"
BSC6900wb.Worksheets(BSC6900ws3).Cells(lastrow + countL18, 39) = "0"
BSC6900wb.Worksheets(BSC6900ws3).Cells(lastrow + countL18, 40) =
BSC6900wb.Worksheets(BSC6900ws3).Cells(ODD_L18 - L18num, 40).Value
BSC6900wb.Worksheets(BSC6900ws3).Cells(lastrow + countL18, 41) = "NOT_CFG"
BSC6900wb.Worksheets(BSC6900ws3).Cells(lastrow + countL18, 42) = "" 'EMPTY
BSC6900wb.Worksheets(BSC6900ws3).Cells(lastrow + countL18, 43) = "" 'EMPTY
BSC6900wb.Worksheets(BSC6900ws3).Cells(lastrow + countL18, 44) =
BSC6900wb.Worksheets(BSC6900ws3).Cells(ODD_L18 - L18num, 44).Value
BSC6900wb.Worksheets(BSC6900ws3).Cells(lastrow + countL18, 45) = "6"
BSC6900wb.Worksheets(BSC6900ws3).Cells(lastrow + countL18, 46) = "NOT_CFG"
BSC6900wb.Worksheets(BSC6900ws3).Cells(lastrow + countL18, 47) = "" 'EMPTY
BSC6900wb.Worksheets(BSC6900ws3).Cells(lastrow + countL18, 48) = "NOT_CFG"
BSC6900wb.Worksheets(BSC6900ws3).Cells(lastrow + countL18, 49) = "" 'EMPTY
BSC6900wb.Worksheets(BSC6900ws3).Cells(lastrow + countL18, 50) = "BOOLEAN_FALSE"
BSC6900wb.Worksheets(BSC6900ws3).Cells(lastrow + countL18, 51) = "" 'EMPTY
BSC6900wb.Worksheets(BSC6900ws3).Cells(lastrow + countL18, 52) = "2T2R"
BSC6900wb.Worksheets(BSC6900ws3).Cells(lastrow + countL18, 53) = "CRS_PORT_2"
BSC6900wb.Worksheets(BSC6900ws3).Cells(lastrow + countL18, 54) = "" 'EMPTY
BSC6900wb.Worksheets(BSC6900ws3).Cells(lastrow + countL18, 55) =
"NO_COMPRESSION"
BSC6900wb.Worksheets(BSC6900ws3).Cells(lastrow + countL18, 56) = "0"

'column 57
'LTE cell template
allLdig = BSC6900wb.Worksheets(BSC6900ws3).Cells(lastrow + countL18, 3).Value
If Mid(allLdig, 7, 2) = 18 Then 'L1800
    BSC6900wb.Worksheets(BSC6900ws3).Cells(lastrow + countL18, 57) =
ThisWorkbook.Worksheets("Data").Range("B2").Value
ElseIf Mid(allLdig, 7, 2) = 21 Then 'L2100
    BSC6900wb.Worksheets(BSC6900ws3).Cells(lastrow + countL18, 57) =
ThisWorkbook.Worksheets("Data").Range("B3").Value
ElseIf Mid(allLdig, 7, 2) = 9 Then 'L900
    BSC6900wb.Worksheets(BSC6900ws3).Cells(lastrow + countL18, 57) =
ThisWorkbook.Worksheets("Data").Range("B4").Value
End If

BSC6900wb.Worksheets(BSC6900ws3).Cells(lastrow + countL18, 58) =
CStr(BSC6900wb.Worksheets(BSC6900ws3).Cells(ODD_L18 - L18num, 2).Value) + 20
BSC6900wb.Worksheets(BSC6900ws3).Cells(lastrow + countL18, 59) = "BOOLEAN_TRUE"

```

```

BSC6900wb.Worksheets(BSC6900ws3).Cells(lastrow + countL18, 60) = "DEFAULTMODE"
BSC6900wb.Worksheets(BSC6900ws3).Cells(lastrow + countL18, 61) =
CStr(BSC6900wb.Worksheets(BSC6900ws3).Cells(ODD_L18 - L18num, 2).Value) + 20
BSC6900wb.Worksheets(BSC6900ws3).Cells(lastrow + countL18, 62) = "OFF"
BSC6900wb.Worksheets(BSC6900ws3).Cells(lastrow + countL18, 63) = "ms40"
BSC6900wb.Worksheets(BSC6900ws3).Cells(lastrow + countL18, 64) =
CStr(BSC6900wb.Worksheets(BSC6900ws3).Cells(ODD_L18 - L18num, 2).Value) + 20
BSC6900wb.Worksheets(BSC6900ws3).Cells(lastrow + countL18, 65) = "OFF"
BSC6900wb.Worksheets(BSC6900ws3).Cells(lastrow + countL18, 66) = "DB_3_P_A"
BSC6900wb.Worksheets(BSC6900ws3).Cells(lastrow + countL18, 67) =
CStr(BSC6900wb.Worksheets(BSC6900ws3).Cells(ODD_L18 - L18num, 2).Value) + 20
BSC6900wb.Worksheets(BSC6900ws3).Cells(lastrow + countL18, 68) = "MOD3"
BSC6900wb.Worksheets(BSC6900ws3).Cells(lastrow + countL18, 69) =
CStr(BSC6900wb.Worksheets(BSC6900ws3).Cells(ODD_L18 - L18num, 2).Value) + 20

```

```
'column 71
```

```
'sector equipment
```

```
sectorL = BSC6900wb.Worksheets(BSC6900ws3).Cells(ODD_L18 - L18num, 2).Value
```

```
If Right(sectorL, 1) = 1 Then
```

```
    BSC6900wb.Worksheets(BSC6900ws3).Cells(lastrow + countL18, 70) =
(CStr(Right(sectorL, 1)) - 1 + 50)
```

```
    ElseIf Right(sectorL, 1) = 2 Then
```

```
        BSC6900wb.Worksheets(BSC6900ws3).Cells(lastrow + countL18, 70) =
(CStr(Right(sectorL, 1)) - 1 + 50)
```

```
    ElseIf Right(sectorL, 1) = 3 Then
```

```
        BSC6900wb.Worksheets(BSC6900ws3).Cells(lastrow + countL18, 70) =
(CStr(Right(sectorL, 1)) - 1 + 50)
```

```
    ElseIf Right(sectorL, 1) = 4 Then
```

```
        BSC6900wb.Worksheets(BSC6900ws3).Cells(lastrow + countL18, 70) =
(CStr(Right(sectorL, 1)) - 1 + 50)
```

```
    ElseIf Right(sectorL, 1) = 5 Then
```

```
        BSC6900wb.Worksheets(BSC6900ws3).Cells(lastrow + countL18, 70) =
(CStr(Right(sectorL, 1)) - 1 + 50)
```

```
    ElseIf Right(sectorL, 1) = 6 Then
```

```
        BSC6900wb.Worksheets(BSC6900ws3).Cells(lastrow + countL18, 70) =
(CStr(Right(sectorL, 1)) - 1 + 50)
```

```
End If
```

```
'column 71
```

```
'band group
```

```
If ur < 6 Then
```

```
    BSC6900wb.Worksheets(BSC6900ws3).Cells(lastrow + countL18, 71) = allgroup(ur2)
```

```
    ur = ur + 1
```

```
    If ur = 6 Then
```

```

        ur2 = ur2 + 1
        ur = 0
    End If
End If

countL18 = countL18 + 1
ODD_L18 = ODD_L18 + 1
Loop
'set gridlines
BSC6900wb.Worksheets(BSC6900ws3).Activate
ActiveSheet.UsedRange.Select
Selection.Borders(xlDiagonalDown).LineStyle = xlNone
Selection.Borders(xlDiagonalUp).LineStyle = xlNone
With Selection.Borders(xlEdgeLeft)
    .LineStyle = xlContinuous
    .ColorIndex = 0
    .TintAndShade = 0
    .Weight = xlThin
End With
With Selection.Borders(xlEdgeTop)
    .LineStyle = xlContinuous
    .ColorIndex = 0
    .TintAndShade = 0
    .Weight = xlThin
End With
With Selection.Borders(xlEdgeBottom)
    .LineStyle = xlContinuous
    .ColorIndex = 0
    .TintAndShade = 0
    .Weight = xlThin
End With
With Selection.Borders(xlEdgeRight)
    .LineStyle = xlContinuous
    .ColorIndex = 0
    .TintAndShade = 0
    .Weight = xlThin
End With
With Selection.Borders(xlInsideVertical)
    .LineStyle = xlContinuous
    .ColorIndex = 0
    .TintAndShade = 0
    .Weight = xlThin
End With
With Selection.Borders(xlInsideHorizontal)
    .LineStyle = xlContinuous

```

```
.ColorIndex=0  
.TintAndShade=0  
.Weight=xlThin  
End With  
End Sub
```

4. Code VBA : Module6

```
Sub BSC6900remain()
```

```
'----- Multicarrier Cell Group -----'
```

```
BSC6900ws4 = "Multicarrier Cell Group"
```

```
BSC6900wb.Worksheets(BSC6900ws4).Rows("3:" & Rows.Count).ClearContents  
BSC6900wb.Worksheets(BSC6900ws4).Rows("3:" & Rows.Count).Interior.Color = xlNone  
BSC6900wb.Worksheets(BSC6900ws4).Cells.HorizontalAlignment = xlCenter
```

```
'set font and size
```

```
With BSC6900wb.Worksheets(BSC6900ws4)  
    .Cells.Font.Name = "Arial"  
    .Cells.Font.Size = 10  
End With
```

```
Dim number, u1, u2, u3 As Integer  
number = 0
```

```
Dim texts, SectorQ(0 To 50), duplicateSectorQ(0 To 50) As String  
Do While number < localcellallnum  
    SectorQ(number) = CStr(Right(localcellall(number), 1)) - 1 + 20  
    number = number + 1  
Loop
```

```
u1 = 0
```

```
u3 = 0
```

```
Do While u1 < number
```

```
    texts = "none"
```

```
    u2 = 0
```

```
    Do While u2 < 50
```

```
        If SectorQ(u1) = duplicateSectorQ(u2) Then
```

```
            texts = "found"
```

```
        End If
```

```
    u2 = u2 + 1
```

```
    Loop
```

```
    If texts = "none" Then
```

```
        duplicateSectorQ(u3) = SectorQ(u1)
```

```
        u3 = u3 + 1
```

```
    End If
```

```
    u1 = u1 + 1
```

```
Loop
```

```
Dim e1, e2, e3, d1, aa1, bb1, cc1 As Integer
```

```
Dim countGroup(0 To 50) As String
```

```
e1 = 0
```

```
Do While e1 < u3
```

```
    e3 = 0
```

```
    e2 = 0
```

```
    Do While e2 < number
```

```
        If duplicateSectorQ(e1) = SectorQ(e2) Then
```

```
            e3 = e3 + 1
```

```
        End If
```

```
    e2 = e2 + 1
```

```
    Loop
```

```
    countGroup(e1) = e3
```

```
e1 = e1 + 1
```

```
Loop
```

```
e1 = 0
```

```
Do While e1 < u3
```

```
    If countGroup(e1) > 1 Then
```

```
        d1 = 0
```

```
        cc1 = ""
```

```
        bb1 = 0
```

```
        aa1 = ""
```

```
        Do While d1 < localcellallnum
```

```
            If duplicateSectorQ(e1) = SectorQ(d1) Then
```

```
                If bb1 = 0 Then
```

```
                    cc1 = ""
```

```
                Else
```

```
                    cc1 = ";"
```

```
                End If
```

```
                aa1 = aa1 & cc1 & localcellall(d1)
```

```
                bb1 = bb1 + 1
```

```
            End If
```

```
        d1 = d1 + 1
```

```
        Loop
```

```
        BSC6900wb.Worksheets(BSC6900ws4).Cells(e1 + 3, 1) =
```

```
BSC6900wb.Worksheets(BSC6900ws2).Cells(e1 + 3, 1)
```

```
        BSC6900wb.Worksheets(BSC6900ws4).Cells(e1 + 3, 2) = duplicateSectorQ(e1)
```

```
        BSC6900wb.Worksheets(BSC6900ws4).Cells(e1 + 3, 3) = "HSDPA"
```

```
        BSC6900wb.Worksheets(BSC6900ws4).Cells(e1 + 3, 4) = aa1
```

```
    End If
```

```
e1 = e1 + 1
```

```
Loop
```

```
'set gridlines
```

```

BSC6900wb.Worksheets(BSC6900ws4).Activate
ActiveSheet.UsedRange.Select
Selection.Borders(xlDiagonalDown).LineStyle = xlNone
Selection.Borders(xlDiagonalUp).LineStyle = xlNone
With Selection.Borders(xlEdgeLeft)
    .LineStyle=xlContinuous
    .ColorIndex=0
    .TintAndShade=0
    .Weight = xlThin
End With
With Selection.Borders(xlEdgeTop)
    .LineStyle=xlContinuous
    .ColorIndex=0
    .TintAndShade=0
    .Weight = xlThin
End With
With Selection.Borders(xlEdgeBottom)
    .LineStyle=xlContinuous
    .ColorIndex=0
    .TintAndShade=0
    .Weight = xlThin
End With
With Selection.Borders(xlEdgeRight)
    .LineStyle=xlContinuous
    .ColorIndex=0
    .TintAndShade=0
    .Weight = xlThin
End With
With Selection.Borders(xlInsideVertical)
    .LineStyle=xlContinuous
    .ColorIndex=0
    .TintAndShade=0
    .Weight = xlThin
End With
With Selection.Borders(xlInsideHorizontal)
    .LineStyle=xlContinuous
    .ColorIndex=0
    .TintAndShade=0
    .Weight = xlThin
End With

```

----- InterRatHoUtranGroup -----

BSC6900ws5 = "InterRatHoUtranGroup"

```
BSC6900wb.Worksheets(BSC6900ws5).Rows("3:" & Rows.Count).ClearContents
BSC6900wb.Worksheets(BSC6900ws5).Rows("3:" & Rows.Count).Interior.Color = xlNone
BSC6900wb.Worksheets(BSC6900ws5).Cells.HorizontalAlignment = xlCenter
```

```
'set font and size
```

```
With BSC6900wb.Worksheets(BSC6900ws5)
```

```
    .Cells.Font.Name = "Arial"
```

```
    .Cells.Font.Size = 10
```

```
End With
```

```
ODD_allLTE = 3
```

```
rowallLTE = 3
```

```
Ltime = 0
```

```
Do While BSC6900wb.Worksheets(BSC6900ws3).Cells(ODD_allLTE, Val(topicLCol(2))) <> ""
```

```
    Ltime = Ltime + 1
```

```
    BSC6900wb.Worksheets(BSC6900ws5).Cells(ODD_allLTE, 1) =
```

```
BSC6900wb.Worksheets(BSC6900ws3).Cells(rowallLTE, 1).Value
```

```
    BSC6900wb.Worksheets(BSC6900ws5).Cells(ODD_allLTE, 2) =
```

```
BSC6900wb.Worksheets(BSC6900ws3).Cells(rowallLTE, 2).Value
```

```
    BSC6900wb.Worksheets(BSC6900ws5).Cells(ODD_allLTE, 3) = "0"
```

```
    BSC6900wb.Worksheets(BSC6900ws5).Cells(ODD_allLTE, 4) = "-28"
```

```
    BSC6900wb.Worksheets(BSC6900ws5).Cells(ODD_allLTE, 5) = "-103"
```

```
    BSC6900wb.Worksheets(BSC6900ws5).Cells(ODD_allLTE, 6) = "2"
```

```
    BSC6900wb.Worksheets(BSC6900ws5).Cells(ODD_allLTE, 7) = "320ms"
```

```
    BSC6900wb.Worksheets(BSC6900ws5).Cells(ODD_allLTE, 8) = "-18"
```

```
    BSC6900wb.Worksheets(BSC6900ws5).Cells(ODD_allLTE, 9) = "-101"
```

```
    rowallLTE = rowallLTE + 1
```

```
    ODD_allLTE = ODD_allLTE + 1
```

```
Loop
```

```
lastR = ODD_allLTE
```

```
countLtime = 0
```

```
Do While countLtime < Ltime
```

```
    BSC6900wb.Worksheets(BSC6900ws5).Cells(lastR + countLtime, 1) =
```

```
BSC6900wb.Worksheets(BSC6900ws5).Cells(ODD_allLTE - Ltime, 1).Value
```

```
    BSC6900wb.Worksheets(BSC6900ws5).Cells(lastR + countLtime, 2) =
```

```
BSC6900wb.Worksheets(BSC6900ws5).Cells(ODD_allLTE - Ltime, 2).Value
```

```
    BSC6900wb.Worksheets(BSC6900ws5).Cells(lastR + countLtime, 3) = "1"
```

```
    BSC6900wb.Worksheets(BSC6900ws5).Cells(lastR + countLtime, 4) = "-28"
```

```
    BSC6900wb.Worksheets(BSC6900ws5).Cells(lastR + countLtime, 5) = "-103"
```

```
    BSC6900wb.Worksheets(BSC6900ws5).Cells(lastR + countLtime, 6) = "2"
```

```
    BSC6900wb.Worksheets(BSC6900ws5).Cells(lastR + countLtime, 7) = "320ms"
```

```
    BSC6900wb.Worksheets(BSC6900ws5).Cells(lastR + countLtime, 8) = "-18"
```

```
    BSC6900wb.Worksheets(BSC6900ws5).Cells(lastR + countLtime, 9) = "-101"
```

```
countLtime = countLtime + 1
```

```

ODD_allLTE = ODD_allLTE + 1
Loop

'set gridlines
BSC6900wb.Worksheets(BSC6900ws5).Activate
ActiveSheet.UsedRange.Select
Selection.Borders(xlDiagonalDown).LineStyle = xlNone
Selection.Borders(xlDiagonalUp).LineStyle = xlNone
With Selection.Borders(xlEdgeLeft)
    .LineStyle = xlContinuous
    .ColorIndex = 0
    .TintAndShade = 0
    .Weight = xlThin
End With
With Selection.Borders(xlEdgeTop)
    .LineStyle = xlContinuous
    .ColorIndex = 0
    .TintAndShade = 0
    .Weight = xlThin
End With
With Selection.Borders(xlEdgeBottom)
    .LineStyle = xlContinuous
    .ColorIndex = 0
    .TintAndShade = 0
    .Weight = xlThin
End With
With Selection.Borders(xlEdgeRight)
    .LineStyle = xlContinuous
    .ColorIndex = 0
    .TintAndShade = 0
    .Weight = xlThin
End With
With Selection.Borders(xlInsideVertical)
    .LineStyle = xlContinuous
    .ColorIndex = 0
    .TintAndShade = 0
    .Weight = xlThin
End With
With Selection.Borders(xlInsideHorizontal)
    .LineStyle = xlContinuous
    .ColorIndex = 0
    .TintAndShade = 0
    .Weight = xlThin
End With

```

BSC6900ws6 = "InterRatHoCommGroup"

BSC6900wb.Worksheets(BSC6900ws6).Rows("3:" & Rows.Count).ClearContents
 BSC6900wb.Worksheets(BSC6900ws6).Rows("3:" & Rows.Count).Interior.Color=xlNone
 BSC6900wb.Worksheets(BSC6900ws6).Cells.HorizontalAlignment=xlCenter

'set font and size

With BSC6900wb.Worksheets(BSC6900ws6)

.Cells.Font.Name="Arial"

.Cells.Font.Size=10

End With

ODD_allL=3

rowallL=3

Ltime1=0

Do While BSC6900wb.Worksheets(BSC6900ws3).Cells(ODD_allL, Val(topicLCol(2))) <> ""

Ltime1= Ltime1+1

BSC6900wb.Worksheets(BSC6900ws6).Cells(ODD_allL, 1) =

BSC6900wb.Worksheets(BSC6900ws3).Cells(rowallL, 1).Value

BSC6900wb.Worksheets(BSC6900ws6).Cells(ODD_allL, 2) =

BSC6900wb.Worksheets(BSC6900ws3).Cells(rowallL, 2).Value

BSC6900wb.Worksheets(BSC6900ws6).Cells(ODD_allL, 3) = "0"

BSC6900wb.Worksheets(BSC6900ws6).Cells(ODD_allL, 4) = "2"

BSC6900wb.Worksheets(BSC6900ws6).Cells(ODD_allL, 5) = "640ms"

BSC6900wb.Worksheets(BSC6900ws6).Cells(ODD_allL, 6) = "-111"

BSC6900wb.Worksheets(BSC6900ws6).Cells(ODD_allL, 7) = "-20"

BSC6900wb.Worksheets(BSC6900ws6).Cells(ODD_allL, 8) = "-115"

BSC6900wb.Worksheets(BSC6900ws6).Cells(ODD_allL, 9) = "-24"

BSC6900wb.Worksheets(BSC6900ws6).Cells(ODD_allL, 10) = "-115"

BSC6900wb.Worksheets(BSC6900ws6).Cells(ODD_allL, 11) = "-24"

BSC6900wb.Worksheets(BSC6900ws6).Cells(ODD_allL, 12) = "-115"

BSC6900wb.Worksheets(BSC6900ws6).Cells(ODD_allL, 13) = "-24"

rowallL= rowallL+1

ODD_allL= ODD_allL+1

Loop

lastR1= ODD_allL

countLtime1=0

Do While countLtime1< Ltime1

BSC6900wb.Worksheets(BSC6900ws6).Cells(lastR1+countLtime1, 1) =

BSC6900wb.Worksheets(BSC6900ws6).Cells(ODD_allL - Ltime1, 1).Value

BSC6900wb.Worksheets(BSC6900ws6).Cells(lastR1+countLtime1, 2) =

BSC6900wb.Worksheets(BSC6900ws6).Cells(ODD_allL - Ltime1, 2).Value

```

BSC6900wb.Worksheets(BSC6900ws6).Cells(lastR1+countLtime1, 3) = "1"
BSC6900wb.Worksheets(BSC6900ws6).Cells(lastR1+countLtime1, 4) = "2"
BSC6900wb.Worksheets(BSC6900ws6).Cells(lastR1+countLtime1, 5) = "640ms"
BSC6900wb.Worksheets(BSC6900ws6).Cells(lastR1+countLtime1, 6) = "-111"
BSC6900wb.Worksheets(BSC6900ws6).Cells(lastR1+countLtime1, 7) = "-20"
BSC6900wb.Worksheets(BSC6900ws6).Cells(lastR1+countLtime1, 8) = "-115"
BSC6900wb.Worksheets(BSC6900ws6).Cells(lastR1+countLtime1, 9) = "-24"
BSC6900wb.Worksheets(BSC6900ws6).Cells(lastR1+countLtime1, 10) = "-115"
BSC6900wb.Worksheets(BSC6900ws6).Cells(lastR1+countLtime1, 11) = "-24"
BSC6900wb.Worksheets(BSC6900ws6).Cells(lastR1+countLtime1, 12) = "-115"
BSC6900wb.Worksheets(BSC6900ws6).Cells(lastR1+countLtime1, 13) = "-24"

```

```
countLtime1 = countLtime1 + 1
```

```
ODD_allL = ODD_allL + 1
```

```
Loop
```

```
'set gridlines
```

```
BSC6900wb.Worksheets(BSC6900ws6).Activate
```

```
ActiveSheet.UsedRange.Select
```

```
Selection.Borders(xlDiagonalDown).LineStyle = xlNone
```

```
Selection.Borders(xlDiagonalUp).LineStyle = xlNone
```

```
With Selection.Borders(xlEdgeLeft)
```

```
.LineStyle = xlContinuous
```

```
.ColorIndex = 0
```

```
.TintAndShade = 0
```

```
.Weight = xlThin
```

```
End With
```

```
With Selection.Borders(xlEdgeTop)
```

```
.LineStyle = xlContinuous
```

```
.ColorIndex = 0
```

```
.TintAndShade = 0
```

```
.Weight = xlThin
```

```
End With
```

```
With Selection.Borders(xlEdgeBottom)
```

```
.LineStyle = xlContinuous
```

```
.ColorIndex = 0
```

```
.TintAndShade = 0
```

```
.Weight = xlThin
```

```
End With
```

```
With Selection.Borders(xlEdgeRight)
```

```
.LineStyle = xlContinuous
```

```
.ColorIndex = 0
```

```
.TintAndShade = 0
```

```
.Weight = xlThin
```

```
End With
```

```
With Selection.Borders(xlInsideVertical)
```

```

.LineStyle=xlContinuous
.ColorIndex=0
.TintAndShade=0
.Weight=xlThin
End With
With Selection.Borders(xlInsideHorizontal)
.LineStyle=xlContinuous
.ColorIndex=0
.TintAndShade=0
.Weight=xlThin
End With

```

'-----UtranNFreq-----'

```
BSC6900ws7 = "UtranNFreq"
```

```

BSC6900wb.Worksheets(BSC6900ws7).Rows("3:" & Rows.Count).ClearContents
BSC6900wb.Worksheets(BSC6900ws7).Rows("3:" & Rows.Count).Interior.Color=xlNone
BSC6900wb.Worksheets(BSC6900ws7).Cells.HorizontalAlignment=xlCenter

```

```

'set font and size
With BSC6900wb.Worksheets(BSC6900ws7)
.Cells.Font.Name = "Arial"
.Cells.Font.Size = 10
End With

```

```

ODD_L = 3
rowL = 3
Ltime2 = 0

```

```

Do While BSC6900wb.Worksheets(BSC6900ws3).Cells(ODD_L, Val(topicLCol(2))) <> ""
Ltime2 = Ltime2 + 1
BSC6900wb.Worksheets(BSC6900ws7).Cells(ODD_L, 1) =
BSC6900wb.Worksheets(BSC6900ws3).Cells(rowL, 1).Value
BSC6900wb.Worksheets(BSC6900ws7).Cells(ODD_L, 2) =
BSC6900wb.Worksheets(BSC6900ws3).Cells(rowL, 2).Value
BSC6900wb.Worksheets(BSC6900ws7).Cells(ODD_L, 3) = "10762"
BSC6900wb.Worksheets(BSC6900ws7).Cells(ODD_L, 4) = "Priority_2"
BSC6900wb.Worksheets(BSC6900ws7).Cells(ODD_L, 5) = "Priority_16"
BSC6900wb.Worksheets(BSC6900ws7).Cells(ODD_L, 6) = "CFG"
BSC6900wb.Worksheets(BSC6900ws7).Cells(ODD_L, 7) = "3"
BSC6900wb.Worksheets(BSC6900ws7).Cells(ODD_L, 8) = "3"
BSC6900wb.Worksheets(BSC6900ws7).Cells(ODD_L, 9) = "6"
BSC6900wb.Worksheets(BSC6900ws7).Cells(ODD_L, 10) = "6"
BSC6900wb.Worksheets(BSC6900ws7).Cells(ODD_L, 11) = "-58"
BSC6900wb.Worksheets(BSC6900ws7).Cells(ODD_L, 12) = "Priority_2"

```

```

rowL = rowL + 1
ODD_L = ODD_L + 1
Loop

lastR2 = ODD_L
countLtime2 = 0
reLtime = 0
Do While countLtime2 < Ltime2
    reLtime = reLtime + 1
    BSC6900wb.Worksheets(BSC6900ws7).Cells(lastR2 + countLtime2, 1) =
BSC6900wb.Worksheets(BSC6900ws7).Cells(ODD_L - Ltime2, 1).Value
    BSC6900wb.Worksheets(BSC6900ws7).Cells(lastR2 + countLtime2, 2) =
BSC6900wb.Worksheets(BSC6900ws7).Cells(ODD_L - Ltime2, 2).Value
    BSC6900wb.Worksheets(BSC6900ws7).Cells(lastR2 + countLtime2, 3) = "10788"
    BSC6900wb.Worksheets(BSC6900ws7).Cells(lastR2 + countLtime2, 4) = "Priority_16"
    BSC6900wb.Worksheets(BSC6900ws7).Cells(lastR2 + countLtime2, 5) = "Priority_2"
    BSC6900wb.Worksheets(BSC6900ws7).Cells(lastR2 + countLtime2, 6) = "CFG"
    BSC6900wb.Worksheets(BSC6900ws7).Cells(lastR2 + countLtime2, 7) = "3"
    BSC6900wb.Worksheets(BSC6900ws7).Cells(lastR2 + countLtime2, 8) = "3"
    BSC6900wb.Worksheets(BSC6900ws7).Cells(lastR2 + countLtime2, 9) = "6"
    BSC6900wb.Worksheets(BSC6900ws7).Cells(lastR2 + countLtime2, 10) = "6"
    BSC6900wb.Worksheets(BSC6900ws7).Cells(lastR2 + countLtime2, 11) = "-58"
    BSC6900wb.Worksheets(BSC6900ws7).Cells(lastR2 + countLtime2, 12) = "Priority_16"
countLtime2 = countLtime2 + 1
ODD_L = ODD_L + 1
Loop

lastR3 = ODD_L
countLtime3 = 0
reLtime1 = 0
Do While countLtime3 < reLtime
    reLtime1 = reLtime1 + 1
    BSC6900wb.Worksheets(BSC6900ws7).Cells(lastR3 + countLtime3, 1) =
BSC6900wb.Worksheets(BSC6900ws7).Cells(ODD_L - reLtime, 1).Value
    BSC6900wb.Worksheets(BSC6900ws7).Cells(lastR3 + countLtime3, 2) =
BSC6900wb.Worksheets(BSC6900ws7).Cells(ODD_L - reLtime, 2).Value
    BSC6900wb.Worksheets(BSC6900ws7).Cells(lastR3 + countLtime3, 3) = "10812"
    BSC6900wb.Worksheets(BSC6900ws7).Cells(lastR3 + countLtime3, 4) = "Priority_16"
    BSC6900wb.Worksheets(BSC6900ws7).Cells(lastR3 + countLtime3, 5) = "Priority_2"
    BSC6900wb.Worksheets(BSC6900ws7).Cells(lastR3 + countLtime3, 6) = "CFG"
    BSC6900wb.Worksheets(BSC6900ws7).Cells(lastR3 + countLtime3, 7) = "3"
    BSC6900wb.Worksheets(BSC6900ws7).Cells(lastR3 + countLtime3, 8) = "3"
    BSC6900wb.Worksheets(BSC6900ws7).Cells(lastR3 + countLtime3, 9) = "6"
    BSC6900wb.Worksheets(BSC6900ws7).Cells(lastR3 + countLtime3, 10) = "6"
    BSC6900wb.Worksheets(BSC6900ws7).Cells(lastR3 + countLtime3, 11) = "-58"

```

```

    BSC6900wb.Worksheets(BSC6900ws7).Cells(lastR3+countLtime3, 12) = "Priority_16"
countLtime3= countLtime3+ 1
ODD_L = ODD_L + 1
Loop

lastR4= ODD_L
countLtime4= 0
reLtime2= 0
Do While countLtime4< reLtime1
    reLtime2= reLtime2+ 1
    BSC6900wb.Worksheets(BSC6900ws7).Cells(lastR4+countLtime4, 1) =
BSC6900wb.Worksheets(BSC6900ws7).Cells(ODD_L - reLtime1, 1).Value
    BSC6900wb.Worksheets(BSC6900ws7).Cells(lastR4+countLtime4, 2) =
BSC6900wb.Worksheets(BSC6900ws7).Cells(ODD_L - reLtime1, 2).Value
    BSC6900wb.Worksheets(BSC6900ws7).Cells(lastR4+countLtime4, 3) = "10837"
    BSC6900wb.Worksheets(BSC6900ws7).Cells(lastR4+countLtime4, 4) = "Priority_16"
    BSC6900wb.Worksheets(BSC6900ws7).Cells(lastR4+countLtime4, 5) = "Priority_2"
    BSC6900wb.Worksheets(BSC6900ws7).Cells(lastR4+countLtime4, 6) = "CFG"
    BSC6900wb.Worksheets(BSC6900ws7).Cells(lastR4+countLtime4, 7) = "3"
    BSC6900wb.Worksheets(BSC6900ws7).Cells(lastR4+countLtime4, 8) = "3"
    BSC6900wb.Worksheets(BSC6900ws7).Cells(lastR4+countLtime4, 9) = "6"
    BSC6900wb.Worksheets(BSC6900ws7).Cells(lastR4+countLtime4, 10) = "6"
    BSC6900wb.Worksheets(BSC6900ws7).Cells(lastR4+countLtime4, 11) = "-58"
    BSC6900wb.Worksheets(BSC6900ws7).Cells(lastR4+countLtime4, 12) = "Priority_16"
countLtime4= countLtime4+ 1
ODD_L = ODD_L + 1
Loop

'set gridlines
BSC6900wb.Worksheets(BSC6900ws7).Activate
ActiveSheet.UsedRange.Select
Selection.Borders(xlDiagonalDown).LineStyle = xlNone
Selection.Borders(xlDiagonalUp).LineStyle = xlNone
With Selection.Borders(xlEdgeLeft)
    .LineStyle=xlContinuous
    .ColorIndex=0
    .TintAndShade=0
    .Weight= xlThin
End With
With Selection.Borders(xlEdgeTop)
    .LineStyle=xlContinuous
    .ColorIndex=0
    .TintAndShade=0
    .Weight= xlThin
End With

```

```
With Selection.Borders(xlEdgeBottom)
```

```
    .LineStyle=xlContinuous
```

```
    .ColorIndex=0
```

```
    .TintAndShade=0
```

```
    .Weight = xlThin
```

```
End With
```

```
With Selection.Borders(xlEdgeRight)
```

```
    .LineStyle=xlContinuous
```

```
    .ColorIndex=0
```

```
    .TintAndShade=0
```

```
    .Weight = xlThin
```

```
End With
```

```
With Selection.Borders(xlInsideVertical)
```

```
    .LineStyle=xlContinuous
```

```
    .ColorIndex=0
```

```
    .TintAndShade=0
```

```
    .Weight = xlThin
```

```
End With
```

```
With Selection.Borders(xlInsideHorizontal)
```

```
    .LineStyle=xlContinuous
```

```
    .ColorIndex=0
```

```
    .TintAndShade=0
```

```
    .Weight = xlThin
```

```
End With
```

```
'-----TcelpMapping-----'
```

```
BSC6900ws8 = "TcelpMapping"
```

```
BSC6900wb.Worksheets(BSC6900ws8).Rows("3:" & Rows.Count).ClearContents
```

```
BSC6900wb.Worksheets(BSC6900ws8).Rows("3:" & Rows.Count).Interior.Color=xlNone
```

```
BSC6900wb.Worksheets(BSC6900ws8).Cells.HorizontalAlignment=xlCenter
```

```
'set font and size
```

```
With BSC6900wb.Worksheets(BSC6900ws8)
```

```
    .Cells.Font.Name="Arial"
```

```
    .Cells.Font.Size=10
```

```
End With
```

```
ODD_reL = 3
```

```
countreL = 0
```

```
TCE = 0
```

```
Do While countreL < 13
```

```
    BSC6900wb.Worksheets(BSC6900ws8).Cells(ODD_reL + countreL, 1) =  
BSC6900wb.Worksheets(BSC6900ws7).Cells(ODD_reL + countreL, 1).Value
```

```

BSC6900wb.Worksheets(BSC6900ws8).Cells(ODD_reL + countreL, 2) = TCE
BSC6900wb.Worksheets(BSC6900ws8).Cells(ODD_reL + countreL, 3) = "IPv4"

TCEAddress = BSC6900wb.Worksheets(BSC6900ws1).Cells(3, 52).Value

If CStr(Left(TCEAddress, 11)) = "10.216.197." Then
    cutTCE = CStr(Left(TCEAddress, 11))
    addTCE = 157 + TCE
    allTCE = CStr(cutTCE) + CStr(addTCE)

    BSC6900wb.Worksheets(BSC6900ws8).Cells(ODD_reL + countreL, 4) = allTCE

ElseIf CStr(Left(TCEAddress, 11)) = "10.219.209." Then
    cutTCE = CStr(Left(TCEAddress, 11))
    addTCE = 149 + TCE
    allTCE = CStr(cutTCE) + CStr(addTCE)

    BSC6900wb.Worksheets(BSC6900ws8).Cells(ODD_reL + countreL, 4) = allTCE
End If
TCE = TCE + 1
addTCE = addTCE + 1
countreL = countreL + 1
Loop

'set gridlines
BSC6900wb.Worksheets(BSC6900ws8).Activate
ActiveSheet.UsedRange.Select
Selection.Borders(xlDiagonalDown).LineStyle = xlNone
Selection.Borders(xlDiagonalUp).LineStyle = xlNone
With Selection.Borders(xlEdgeLeft)
    .LineStyle = xlContinuous
    .ColorIndex = 0
    .TintAndShade = 0
    .Weight = xlThin
End With
With Selection.Borders(xlEdgeTop)
    .LineStyle = xlContinuous
    .ColorIndex = 0
    .TintAndShade = 0
    .Weight = xlThin
End With
With Selection.Borders(xlEdgeBottom)
    .LineStyle = xlContinuous
    .ColorIndex = 0
    .TintAndShade = 0

```

```

        .Weight = xlThin
    End With
    With Selection.Borders(xlEdgeRight)
        .LineStyle = xlContinuous
        .ColorIndex = 0
        .TintAndShade = 0
        .Weight = xlThin
    End With
    With Selection.Borders(xlInsideVertical)
        .LineStyle = xlContinuous
        .ColorIndex = 0
        .TintAndShade = 0
        .Weight = xlThin
    End With
    With Selection.Borders(xlInsideHorizontal)
        .LineStyle = xlContinuous
        .ColorIndex = 0
        .TintAndShade = 0
        .Weight = xlThin
    End With

    If coopWB.Worksheets("process").Range("M4") <> "" Or
coopWB.Worksheets("process").Range("N4") <> "" Then
        MsgBox "Error!! Please check your data.", vbExclamation
        BSC6900wb.Close savechanges:=False
        coopWB.Worksheets("process").Activate
    Else
        BSC6900wb.Worksheets("Base Station Transport Data").Activate
        BSC6900wb.SaveAs filename:=planwb.Path & "\NewSiteGenerator\" & _
        BSC6900wb.Worksheets(IP4Poolws1).Cells(3, 54).Value & "_" & Year(Now) & "_" &
Month(Now) & _
        "_" & Day(Now) & "-" & Hour(Now) & "H_" & Minute(Now) & "M_" & Second(Now) &
"S.xlsm"
    End If

End Sub

```