

การศึกษาและจำลองการป้องกันในระบบจำหน่ายโดยอัตโนมัติ

A STUDYING AND MODELLING THE PROTECTION
IN DISTRIBUTION AUTOMATION SYSTEM



นายรัชชัย ธรรมประสิทธิ์

MR. THAWATCHAI THAMMAPRASIT

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรมหาบัณฑิต

สาขาวิชาวิศวกรรมไฟฟ้า

บัณฑิตวิทยาลัย

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

พ.ศ.2541

ISBN 974-622-287-2

ลิขสิทธิ์ของบัณฑิตวิทยาลัย สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**A STUDYING AND MODELLING THE PROTECTION IN
DISTRIBUTION AUTOMATION SYSTEM**



**A THESIS SUBMITTED IN PARTIAL FULFILMENT
OF THE REQUIREMENT FOR THE DEGREE
MASTER OF ENGINEERING IN ELECTRICAL ENGINEERING
SCHOOL OF GRADUATE STUDIES
KING MONGKUT'S INSTTTUE OF TECHNOLOGY LADKRABANG**

1998

ISBN 974-622-287-2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หัวข้อวิทยานิพนธ์

นักศึกษา

อาจารย์ผู้ควบคุมวิทยานิพนธ์

ระดับการศึกษา

พ.ศ.

การศึกษาและจำลองการป้องกันในระบบจำหน่ายโดยอัตโนมัติ

นาย ธวัชชัย ธรรมประสิทธิ์

รศ. มณฑล ลีลาจินดาไกรฤกษ์

วิศวกรรมศาสตรมหาบัณฑิต สาขาวิศวกรรมไฟฟ้า

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

2541

บทคัดย่อ

วิทยานิพนธ์นี้ เป็นการศึกษาและจำลองการป้องกันในระบบจำหน่ายโดยอัตโนมัติ โดยการนำเทคโนโลยีทางด้านคอมพิวเตอร์และด้านการสื่อสารข้อมูลระหว่างคอมพิวเตอร์ (Data Communication) มาประยุกต์ใช้ซึ่งทำให้เกิดการแลกเปลี่ยนข้อมูลหรือคำสั่งระหว่างคอมพิวเตอร์ในศูนย์ควบคุม (Master Station) กับสถานีระยะไกล (Remote Terminal Unit) และจัดเก็บข้อมูลของระบบจำหน่ายให้มีประสิทธิภาพมากยิ่งขึ้น ซึ่งโครงสร้างของระบบการป้องกันที่จำลองนี้ประกอบด้วยส่วนสำคัญ 3 ส่วนคือ ศูนย์ควบคุม (Master Station) สถานีระยะไกล (Remote Terminal Unit) และระบบสื่อสาร (Communication Media) โดยมีเป้าหมายในการจัดการที่อุปกรณ์ในสายป้อนของระบบจำหน่ายและไปสิ้นสุดที่เครื่องวัดของผู้ใช้ไฟ

Thesis Title A studying and Modelling the protection in Distribution Automation System

Student Mr. Thawatchai Thammaprasit

Thesis Adviser Assoc. Prof. Monthon Leelajindakrairerk

Level of Study Master of Engineering in Electrical Engineering
King Mongkut's Institute of Technology Ladkrabang

Year 1998

ABSTRACT

This thesis presents a Studying and modulling the protection automation system by the application of computer technology and data communications. This system results in the interchange of data or commands between the computer in master station and the remote control unit. It also stores information on the distribution system more efficiency. The protection system at modulling structure consists of three main parts: the master station, the remote terminal unit and the communication media, to manage its internal equipment in the feeder of distribution system through the consumer's kilowatt-hour meter.

กิตติกรรมประกาศ

วิทยานิพนธ์ฉบับนี้สำเร็จลงด้วยดีก็เพราะได้รับความอนุเคราะห์และคำแนะนำต่าง ๆ จาก ผศ. มณฑล ลีลาจินดาไกรฤกษ์ , ดร.ชัยวุฒิ จัทรอุทัย , อ. พิเชิต ล้ำยอง , อ. พลผดุง ผดุงกุล รศ.ศุสึ บรรจงจิตร , อ.นิทัศน์ กฤษณะจินดา และ รศ.ประภาส ไพรสุวรรณ ที่กรุณาให้คำแนะนำ และให้ความช่วยเหลือทุกด้านแก่ผู้วิจัยตลอดมา ผู้วิจัยขอขอบพระคุณเป็นอย่างสูง

สุดท้ายนี้ผู้วิจัยขอขอบคุณ คุณวันชัย จงวิริยะเจริญชัย , คุณสมชาย แซ่มสาคร , คุณวนิดา ชูสังข์ และสถานีจ่ายไฟนวนคร การไฟฟ้าส่วนภูมิภาค ที่ได้ให้คำแนะนำและให้ความช่วยเหลือจนวิทยานิพนธ์สำเร็จลงด้วยดี



สารบัญ

	หน้า
บทคัดย่อภาษาไทย.....	I
บทคัดย่อภาษาอังกฤษ.....	II
กิตติกรรมประกาศ.....	III
สารบัญ.....	IV
สารบัญตาราง.....	VI
สารบัญภาพ.....	VIII
บทที่	
1 บทนำ.....	1
1.1 แนวความคิดของวิทยานิพนธ์.....	1
1.2 วัตถุประสงค์ของวิทยานิพนธ์.....	2
1.3 ประโยชน์ที่ได้รับจากวิทยานิพนธ์.....	2
1.4 ขอบเขตของวิทยานิพนธ์.....	2
2 ระบบการควบคุมการส่งจ่ายกำลังไฟฟ้า.....	3
2.1 ความรู้เบื้องต้นเกี่ยวกับระบบควบคุมการส่งกำลังไฟฟ้า.....	3
2.2 ระบบส่งจ่ายกำลังไฟฟ้าอัตโนมัติคุณภาพสูง.....	4
2.3 ระบบส่งจ่ายอัตโนมัติโดยใช้การสื่อสาร.....	5
2.4 การทำงานของอุปกรณ์ป้องกัน.....	13
2.5 การทำงานของAutomatic Reclosing Sectionlizer.....	18
2.6 เหตุผลในการเลือกวงจรควบคุมแต่ละชนิด.....	29
2.7 การแยกฟอลต์ออกจากระบบจำหน่าย.....	30
3 ไมโครคอมพิวเตอร์กับการอินเตอร์เฟส.....	33
3.1 หลักการเบื้องต้นของ 8255.....	33
3.2 อุปกรณ์ต่าง ๆ ที่ช่วยในการตีโคตสัญญาณบน IBM.....	42
3.3 การต่อเชื่อม 8255 กับ IBM PC/AT.....	44
3.4 ความรู้ทั่วไปเกี่ยวกับภาษาซี.....	49
3.5 การเขียนโปรแกรมด้วยกราฟฟิกเทอร์โบซี.....	50

สารบัญ (ต่อ)

บทที่	หน้า
3.6	ฟังก์ชันที่ใช้ในการเขียนโปรแกรม..... 54
3.7	คำสั่งควบคุมโปรแกรมภาษาซี..... 58
3.8	โปรแกรมสำหรับทดสอบ Hard ware..... 66
3.9	แบบจำลองการป้องกันและตรวจจับฟอลท์ในระบบจำหน่าย..... 70
3.10	การหาค่ากระแสฟอลท์และแรงดันตกที่จุดต่าง ๆ ในระบบจำหน่าย.. 72
4	โปรแกรมและผลการตรวจจับฟอลท์ในระบบจำหน่าย..... 74
4.1	ขั้นตอนการใช้โปรแกรมร่วมกับชุดจำลอง..... 74
4.2	ผลการทดสอบจากโปรแกรม..... 83
5	สรุปผลการวิจัยและข้อเสนอแนะ..... 92
5.1	สรุปผลการวิจัย..... 92
4.2	ข้อเสนอแนะ..... 93
บรรณานุกรม.....	94
ภาคผนวก	95
ก โปรแกรมที่ใช้ในการตรวจจับฟอลท์.....	96
ข ผลงานที่ลงตีพิมพ์.....	150
ประวัติผู้เขียน.....	166

สารบัญตาราง

ตารางที่	หน้า
1 แสดง characteristic ของ line senser.....	10
2 แสดง characteristic SF6 Gas.....	13
3 แสดงสัญญาณแต่ละพอลต์ของ IC 8255.....	35
4 แสดงหน้าที่ของพอร์ต์ C เป็นสัญญาณ Handshake เมื่อ 8255ทำงานที่โหมด 1.	38
5 แสดงหน้าที่ของพอร์ต์ C ในโหมด 2.....	41
6 แสดง output ของ IC 74LS138.....	44
7 แสดงระบบกราฟฟิกที่ใช้กับเทอร์โบซี.....	50
8 แสดงชื่อและค่าคงที่ของไดรเวอร์ต่าง ๆ.....	52
9 แสดงไดรเวอร์โหมดค่าคงที่และความละเอียดของระบบกราฟฟิก.....	53
10 แสดงสีต่าง ๆ ที่มีในเทอร์โบซี.....	56
11 แสดงลักษณะของเส้นที่มีใช้ในเทอร์โบซี.....	57
12 แสดงขนาดความกว้างของเส้น.....	58
13 แสดงผลการเกิด Three phase fault ชั้นที่ Feeder 1 ที่ระยะทาง 5 km.....	83
14 แสดงผลการเกิด Three phase fault ชั้นที่ Feeder 1 ที่ระยะทาง 10 km.....	84
15 แสดงผลการเกิด Three phase fault ชั้นที่ Feeder 1 ที่ระยะทาง 15 km.....	84
16 แสดงผลการเกิด Three phase fault ชั้นที่ Feeder 1 ที่ระยะทาง 20 km.....	85
17 แสดงผลการเกิด Three phase fault ชั้นที่ Feeder 1 ที่ระยะทาง 25 km.....	85
18 แสดงผลการเกิด Single line to ground fault ชั้นที่ Feeder 2 ที่ระยะทาง 5 km.....	86
19 แสดงผลการเกิด Single line to ground fault ชั้นที่ Feeder 2 ที่ระยะทาง 10 km.....	86
20 แสดงผลการเกิด Single line to ground fault ชั้นที่ Feeder 2 ที่ระยะทาง 15 km.....	87
21 แสดงผลการเกิด Single line to ground fault ชั้นที่ Feeder 2 ที่ระยะทาง 20 km.....	87

VI

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญตาราง (ต่อ)

หน้า

22 แสดงผลการเกิด Single line to ground fault ชั้นที่ Feeder 2 ที่ระยะทาง 25 km.....	88
--	----



VII

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญภาพ

ภาพที่	หน้า
1 แสดง Isolation Pocedure in a Remote-Control System	5
2 แสดงโครงสร้างของ Non-Interruption System.....	6
3 แสดงลักษณะการทำงานของ Master Station.....	7
4 แสดงส่วนต่าง ๆ ของระบบสื่อสารเบื้องต้น.....	8
5 แสดงลักษณะของ remote terminal unit.....	8
6 แสดงส่วนประกอบของชุด remote terminal unit.....	9
7 แสดงรูปร่างของ remote terminal unit.....	9
8 แสดงการติดตั้ง line senser ในระบบจำหน่าย.....	10
9 แสดง block diagram ของ line senser.....	11
10 แสดงลักษณะรูปร่างของ automatic reclosing sectionlizer (SF6).....	12
11 แสดง arc-extingxishing meachanism.....	12
12 แสดงการติดตั้ง automatic reclosing sectionlizer.....	14
13 แสดงเวลาที่ใช้ในการทำงานของ sectionlizer.....	15
14 แสดงการทำงานของวงจรควบคุม sectionlizer แบบ VT-1.....	17
15 แสดงการทำงานของวงจร sectionlizer แบบ VT-2.....	17
16 แสดงการทำงานของวงจรควบคุม sectionlizer แบบ O-T.....	18
17 แสดงการติดตั้ง Automatic reclosing sectionlizer ที่สมบูรณ์.....	19
18 แสดงการทำงานของ circuit breaker ที่กำหนด.....	19
19 แสดงลำดับขั้นตอนการทำงานของ circuit breaker ที่เพิ่มขึ้นเมื่อใช้กับ automatic reclosing sectionlizer.....	20
20 แสดงการทำงานของ automatic reclosing sectionlizer A,B กรณี ไม่มี Z-time.....	21
21 แสดงการทำงานของ autimatic reclosing sectionlizer เมื่อเกิดฟอลท์ ใน ส่วนที่ 1.....	21
22 แสดงการทำงานของ automatic reclosing sectionlizer เมื่อเกิดฟอลท์ ใน ส่วนที่1 ที่เวลาต่าง ๆ.....	22
23 แสดงการทำงานของ automatic reclosing sectionlizer เมื่อเกิดฟอลท์ ใน ส่วนที่ 2.....	23

สารบัญภาพ (ต่อ)

	หน้า
24 แสดงการทำงานของ automatic reclosing sectionlizer เมื่อเกิดฟอลท์ ในส่วนที่ 2 ที่เวลาต่าง ๆ.....	24
25 แสดงการทำงานของ automatic reclosing sectionlizer เมื่อเกิดฟอลท์ ในส่วนที่ 3.....	26
26 แสดงขั้นตอนการทำงานของ automatic reclosing sectionlizer เมื่อเกิดฟอลท์ ในส่วนที่ 3 ช่วงเวลาต่าง ๆ.....	27
27 แสดงการทำงานของ automatic reclosing sectionlizer ในกรณี E (O-T) ไม่มี Z-time.....	29
28 แสดง Isolation procedure on a ground fault.....	31
29 แสดงการ closing และ opening.....	31
30 แสดง Isolation procedure on a short circuit fault.....	32
31 แสดง diagram ของ IC 8255.....	33
32 แสดงตำแหน่งการวางขาของ 8255.....	34
33 แสดงลักษณะการใช้งานของโหมด 0.....	36
34 แสดงโครงสร้างของการจัดสัญญาณ handshake.....	38
35 แสดงการต่อ 8255 ในโหมด.....	40
36 แสดงผังเวลารับ/ส่งข้อมูลแบบ handshake.....	40
37 แสดงโครงสร้างของพอร์ต A ที่ทำงานแบบพอร์ต 2 ทิศทาง.....	42
38 แสดงการจัดขาและตาราง output ของ IC 74LS688.....	43
39 แสดงการจัดขาของ ICLS138.....	44
40 แสดงวงจรตีโค้ดเดอร์.....	46
41 แสดงวงจรที่สมบูรณ์ของแผ่นวงจรอินเทอร์เฟส ด้วย 8255 บนเครื่อง IBM PC.....	48
42 แสดงวงจรแปลงสัญญาณอนาล็อกเป็นดิจิตอล.....	49
43 แสดง single line diagram ของระบบส่งจ่ายกำลังไฟฟ้า.....	70
44 แสดงวิธีการติดตั้ง Automatic Reclosing Sectionlizer.....	71

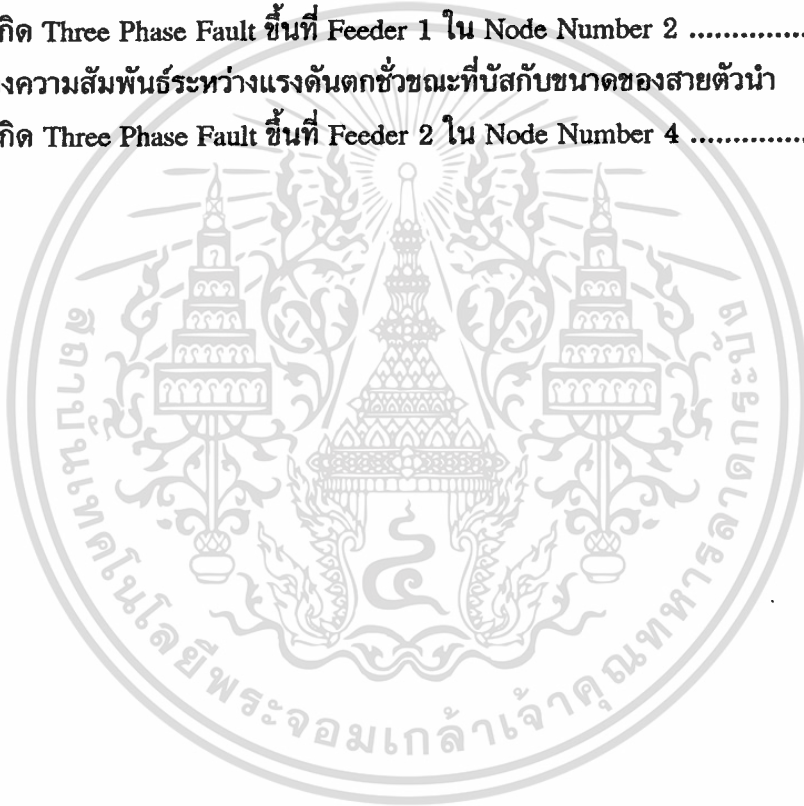
สารบัญภาพ (ต่อ)

	หน้า
45 แสดงการทำงานระหว่างตัวควบคุมกับศูนย์ควบคุม.....	71
46 แสดงการ Single Line Diagram ของ Impedance ในระบบจากแหล่งจ่าย ถึงจุดเกิดฟอลท์.....	72
47 แสดง Flow Chart การทำงานของโปรแกรม.....	74
48 แสดงแผนผังของระบบจำหน่ายที่มี 2 Feeder.....	76
49 แสดงรูปแบบการป้อนข้อมูลลงใน input data.....	76
50 แสดงรูปแบบการป้อนข้อมูลใน feeder 1 และ feeder 2.....	77
51 แสดงรูปแบบการค้นหา Conductor code.....	77
52 แสดงข้อมูลที่กำหนดลงใน feeder 1 และ feeder 2.....	78
53 แสดงการจ่ายไฟในสภาวะปกติของทั้ง 2 Feeder จากชุดจำลองที่สร้างขึ้น.....	79
54 แสดงการเกิดฟอลท์แบบ three phase fault ใน Feeder 1 ที่ Node Number 2.....	79
55 แสดงบริเวณกระแสไฟฟ้าขัดข้องเมื่อเกิด Three Phase Fault จาก ชุดจำลองที่สร้างขึ้น.....	80
56 แสดงการเกิดฟอลท์แบบ single line to ground fault ใน Feeder 2 ที่ Node Number 4.....	81
57 แสดงการเกิดฟอลท์แบบ single line to ground fault และ three phase fault ขึ้นพร้อมกัน.....	81
58 แสดงการเกิดกระแสไฟฟ้าขัดข้องทั้ง 2 feeder จากชุดจำลองที่สร้างขึ้น.....	82
59 แสดงความสัมพันธ์ระหว่างกระแสฟอลท์กับระยะทางเมื่อเกิด Three Phase - Fault ขึ้นที่ Feeder 1 ใน Node Number 2 ของสายตัวนำขนาดต่าง ๆ.....	89
60 แสดงความสัมพันธ์ระหว่างกระแสฟอลท์กับระยะทางเมื่อเกิด Single line to ground Fault ขึ้นที่ Feeder 2 ใน Node Number 4 ของสายตัวนำขนาดต่าง ๆ..	89
61 แสดงความสัมพันธ์ระหว่างแรงดันตกกับระยะทางเมื่อเกิด Three Phase - Fault ขึ้นที่ Feeder 1 ใน Node Number 2 ของสายตัวนำขนาดต่าง ๆ.....	90

สารบัญภาพ (ต่อ)

หน้า

62 แสดงความสัมพันธ์ระหว่างแรงดันตกกับระยะทางเมื่อเกิด Singlr line to ground Fault ขึ้นที่ Feeder 2 ใน Node Number 4 ของสายตัวนำขนาดต่าง ๆ...	90
63 แสดงความสัมพันธ์ระหว่างแรงดันตกชั่วขณะที่ยับกับขนาดของสายตัวนำเมื่อเกิด Three Phase Fault ขึ้นที่ Feeder 1 ใน Node Number 2	91
64 แสดงความสัมพันธ์ระหว่างแรงดันตกชั่วขณะที่ยับกับขนาดของสายตัวนำเมื่อเกิด Three Phase Fault ขึ้นที่ Feeder 2 ใน Node Number 4	91



บทที่ 1

บทนำ

1.1 แนวความคิดของวิทยานิพนธ์

ในสภาพสังคมและเศรษฐกิจปัจจุบันมีความเจริญก้าวหน้าอย่างรวดเร็ว ซึ่งทำให้การวางแผนการควบคุมและป้องกันการจ่ายพลังงานไฟฟ้าในระบบจำหน่ายมีความจำเป็นมากขึ้น การจะทำให้ระบบควบคุมและป้องกันเกิดความมั่นคงมากขึ้นในระบบจำหน่ายและเหมาะสมกับสภาพความต้องการของผู้ใช้ไฟฟ้าก่อให้เกิดประโยชน์สูงสุด จำเป็นต้องมีอุปกรณ์ที่ทันสมัยและมีประสิทธิภาพช่วยในการควบคุม ซึ่งได้มีการพัฒนาโดยประยุกต์เอาคอมพิวเตอร์และอุปกรณ์สื่อสารข้อมูลที่มีประสิทธิภาพสูงมาช่วยควบคุมอุปกรณ์ป้องกัน (Automatic reclosing sectionlizer) ที่ติดตั้งในระบบจำหน่าย โดยการกำหนดระยะเวลาที่ติดตั้งอุปกรณ์ป้องกันให้สามารถบอกสถานที่ของการเกิดฟอลต์ได้ ซึ่งเมื่อเกิดเหตุขัดข้องในระบบจำหน่ายจะใช้เวลาในการตรวจสอบหาจุดที่เกิดฟอลต์และสาเหตุที่ขัดข้องในสายป้อนของระบบจำหน่ายได้สะดวกมากขึ้น

ดังนั้น วิทยานิพนธ์นี้จึงเสนองานวิจัยและทำการพัฒนาระบบการจัดการในสายป้อนของระบบจำหน่าย โดยการสร้างชุดจำลองและใช้ Relay เป็น Automatic reclosing sectionlizer เพื่อแยกส่วนที่เกิดฟอลต์ในสายป้อนออก โดยใช้เทคนิคการบอกตำแหน่งฟอลต์ในระบบจำหน่ายด้วยวิธีการส่งสัญญาณพัลส์ออกไปแล้ววัดสัญญาณที่ตอบสนองกลับมาและประมวลผลออกทางจอภาพของคอมพิวเตอร์ ซึ่งการประมวลผลนั้นจะมีการคำนวณระดับแรงดันตกชั่วขณะ (Voltage - Drip) , กระแสลัดวงจร (Short Circuit Current) ในสภาวะเกิดฟอลต์และรายงานผลทางจอภาพคอมพิวเตอร์เพื่อให้ผู้ควบคุมทราบและทำการ โอนถ่ายโหลดระหว่างสายป้อนเพื่อให้เกิดบริเวณไฟดับในสายป้อนของระบบจำหน่ายน้อยที่สุด

1.2 วัตถุประสงค์ของวิทยานิพนธ์

1. เพื่อศึกษาและพัฒนาระบบการจัดการป้องกันในระบบจำหน่ายให้มีประสิทธิภาพมากยิ่งขึ้น
2. เพื่อนำเทคโนโลยีทางด้านการสื่อสารสมัยใหม่เข้ามาประยุกต์ใช้งานทางด้าน การป้องกันในสายป้อนของระบบจำหน่ายให้มีความสะดวกและรวดเร็วมากยิ่งขึ้น
3. เพื่อศึกษาการทำงานของอุปกรณ์ที่ติดตั้งอยู่ในชุดจำลองของระบบไฟฟ้าให้ทำงานตามเงื่อนไข

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4. เพื่อนำเทคโนโลยีทางด้านคอมพิวเตอร์มาประยุกต์ใช้ในการแสดงผลและประมวลผลการเกิดฟอลท์ทางออกจอภาพของคอมพิวเตอร์

1.3 ประโยชน์ที่ได้รับจากวิทยานิพนธ์

1. สามารถศึกษาการจำกัดเก็บรวบรวมข้อมูลในสายป้อนของระบบจำหน่ายจากชุดจำลองโดยการอ่านสถานะการทำงานของอุปกรณ์ที่ติดตั้งในระบบทั้ง 2 Feeder
2. สามารถนำงานวิจัยในวิทยานิพนธ์นี้ไปใช้ในการศึกษาและควบคุมการทำงานของอุปกรณ์ทุกตัวที่ต้องการควบคุมภายในสายป้อนของระบบจำหน่ายนั้นได้
3. สามารถช่วยลดการขัดข้องในระบบจำหน่ายทั้งในส่วนของเวลา และพื้นที่ที่เกิดการขัดข้องในสายป้อนนั้น ๆ อันได้แก่
 - 3.1 ตัดบริเวณที่เกิดฟอลท์ออกจากระบบจำหน่ายได้ในเวลาที่เหมาะสม
 - 3.2 สนับสนุนทางด้านข้อมูลแก่งานบริการแก้ไขไฟฟ้าขัดข้องให้ผู้ใช้ไฟฟ้าทราบ
 - 3.3 ปลด-ดับสวิทช์โอนถ่ายโหลดในสายป้อนโดยสั่งการจากคอมพิวเตอร์
4. ซอฟต์แวร์คอมพิวเตอร์สามารถนำไปใช้ในการค้นคว้าและวิจัยระบบจำหน่ายได้ต่อไป

1.4 ขอบเขตของวิทยานิพนธ์

ในวิทยานิพนธ์นี้เป็นการนำเสนอการป้องกันในระบบจำหน่ายโดยการสร้างชุดจำลอง (Simulator modelling) และนำเทคโนโลยีทางด้านคอมพิวเตอร์เข้ามาช่วยควบคุมในระบบ ซึ่งได้มีการวิจัยโดยการเขียนซอฟต์แวร์เพื่อติดต่อกับผู้ใช้ในรูปแบบกราฟิก(Graphical) โดยซอฟต์แวร์ที่เขียนขึ้นนี้สามารถ

1. แสดงตำแหน่งที่เกิดฟอลท์ในสายป้อนของระบบจำหน่ายด้วยดีต่าง ๆ
2. แสดงลักษณะการเกิดฟอลท์และระยะทางของสายป้อนในช่วงต่าง ๆ ได้
3. แสดงกระแสลัดวงจร(Short Circuit Current)และแรงดันตก (Voltage Drip)ขณะเกิดฟอลท์แบบ Three Phase Fault และ Single Line to Ground Fault ที่เกิดขึ้นในระบบจำหน่าย
4. แสดงพิกัดของโหลดในสายป้อนนั้น ๆ

บทที่ 2

ระบบการควบคุมการส่งจ่ายกำลังไฟฟ้า

2.1 ความรู้เบื้องต้นเกี่ยวกับระบบควบคุมการส่งจ่ายกำลังไฟฟ้า

ปัจจุบันการใช้พลังงานในประเทศเพิ่มขึ้นค่อนข้างรวดเร็ว โดยเฉพาะอย่างยิ่งพลังงานไฟฟ้าซึ่งทำให้ต้องขยายวงจรรบบจำหน่ายไฟฟ้าและระบบสายส่งกำลังรวมถึงการก่อสร้างสถานีเปลี่ยนแรงดันและสถานีควบคุมการจ่ายไฟมากขึ้น ซึ่งทำให้ระบบ Network ของระบบจำหน่ายและสายส่งมีความซับซ้อนมากยิ่งขึ้น การพิจารณานำเอาระบบ Centralized control มาควบคุมการจ่ายไฟจะช่วยเพิ่มประสิทธิภาพในการจ่ายไฟให้เป็นไปอย่างมีประสิทธิภาพสูงขึ้น

ระบบ Centralized control ได้ถูกนำมาใช้ในการควบคุมการผลิตและการจ่ายไฟโดยเรียกระบบดังกล่าวว่า Supervisory Control and Data Acquisition ระบบนี้จะทำหน้าที่ตรวจสอบการทำงานของเซอร์กิตเบรกเกอร์, โหลด, หม้อแปลงและค่าที่วัดได้ทางไฟฟ้า นอกจากนั้นยังสามารถควบคุมอุปกรณ์ต่าง ๆ ภายในสถานีเปลี่ยนแรงดันและอุปกรณ์ในระบบจำหน่ายได้อีกด้วย ส่วนประกอบของระบบ Power System control สามารถแยกออกเป็นส่วน ๆ ตามหน้าที่การทำงานได้ดังนี้

2.1.1 Remote Terminal Unit (RTU) โครงสร้างจะถูกออกแบบเป็นระบบ Multi-microcomputer ซึ่งจะมีความแน่นอนในการแลกเปลี่ยนข้อมูลระหว่างสถานีเปลี่ยนแรงดันหรือสถานีผลิตถึงศูนย์ควบคุมได้สูงและมีความสามารถในการทำงานได้เร็ว จึงทำให้สามารถตรวจสอบ Sequence of Events ได้ละเอียดภายในเวลาเพียง 2-3 ms เท่านั้น

2.1.2 Computer System เพื่อให้ระบบมีความมั่นคงสูง ระบบคอมพิวเตอร์ควรเป็นระบบ Dual Computer และเพื่อให้ระบบสมบูรณ์ยิ่งขึ้น จะต้องมีคอมพิวเตอร์สำหรับทดสอบและการฝึกอบรมและเป็น Back-up สำหรับงานบางส่วนด้วย

2.1.3 Software โปรแกรมเหล่านี้จะถูกออกแบบมาใช้ทั่ว ๆ ไป ซึ่งโปรแกรมที่ดีจะต้องมีคุณสมบัติดังนี้ ใช้ High level programming language การ link ระหว่าง Module จะต้องทำโดยผ่าน Softbus และ ไม่ขึ้นกับ Hardware ของคอมพิวเตอร์และต้องง่ายต่อการเพิ่ม Application program ในอนาคตตลอดจนความสะดวกต่อการแสดง diagnostics ทางระบบ on-line ด้วยจากที่ได้กล่าวมาแล้วระบบ Power System Control ไม่ได้ขึ้นอยู่กับระบบควบคุมที่ดีและอุปกรณ์ที่มีประสิทธิภาพสูงเท่านั้น ยังมีส่วนประกอบเสริมที่ทำให้ประสิทธิภาพสูงขึ้นหรือลดลงได้ เช่นการออกแบบภายในทางด้านสถาปัตยกรรม เป็นต้น

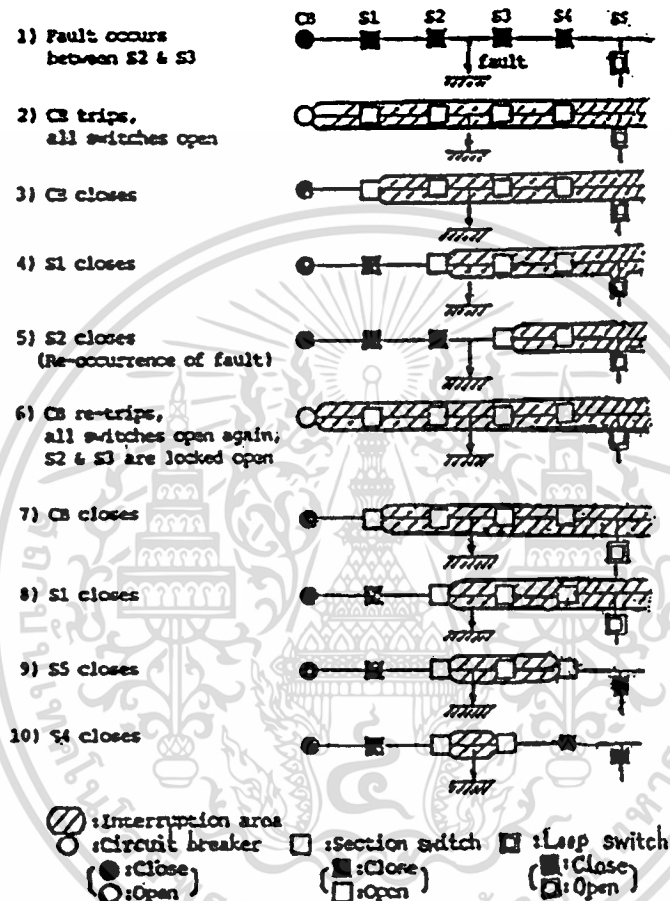
2.2 ระบบส่งจ่ายกำลังไฟฟ้าอัตโนมัติคุณภาพสูง

ระบบส่งจ่ายกำลังไฟฟ้า 3 เฟส 3 สาย ซึ่งใช้แรงดัน 22 kV. ในระบบนี้สวิตช์ตัดคอน (Automatic reclosing sectionlizer) จะทำงานเมื่อเกิดฟอลท์ขึ้นในระบบจำหน่ายไฟฟ้า (Feeder) เพื่อตัดส่วนที่เกิดฟอลท์ออกจากระบบ (Loop Switch) โดยระบบจำหน่ายกำลังไฟฟ้าแบบอัตโนมัติ ถ้าส่วนนั้นจะเป็น ระบบ Time Delay ซึ่งในระบบนี้ ตัวรีเลย์จะเป็นตัวที่ติดตั้งเพื่อตรวจจับฟอลท์ที่เกิดขึ้น (Fault Detecting relay ; FDR) ร่วมกันกับสวิตช์ตัดคอน ซึ่ง Fault Detecting relay แต่ละตัวจะประกอบไปด้วยอุปกรณ์จับเวลา (Built in Time) สองตัว ตัวแรกสำหรับระยะเวลา X- time (9 วินาที) และอีกตัวสำหรับระยะเวลา Y-time (6 วินาที) ระยะเวลา X- time คือช่วงเวลาดังแต่ Fault Detecting relay สั่งให้สวิตช์ตัดคอนเปิดจนกระทั่ง Fault Detecting relay สั่งให้สวิตช์ตัดคอนปิดลง Fault Detecting relay จะทำการ Lock -Out ให้สวิตช์ตัดคอนเปิดทำให้แรงดันลดลงจนกระทั่งเป็นศูนย์ และในระยะ Y-time ตัวสวิตช์ตัดคอน (Circuit Breaker) จะตัดออกอีกครั้งหลังจากที่สวิตช์ตัดคอนปิดลง เมื่อเกิดฟอลท์ขึ้นตัว Circuit Breaker ในสถานีไฟฟ้าย่อยจะทำการ Trip Out และสวิตช์ตัดคอนทั้งหมดก็จะปลดออก ในตอนนี้ Circuit Breaker จะเปิด และในช่วงแรกสวิตช์ตัดคอนจะเปิดในระยะ X-time และสวิตช์ตัดคอนจะปิดไปตามลำดับจนถึงจุดที่เกิดฟอลท์ ตัว Circuit Breaker ก็จะทำการ Trip อีกครั้งและสวิตช์ตัดคอนทั้งหมดก็จะปลดออก ทำให้สวิตช์ตัดคอนตัวที่ใกล้จุดเกิดฟอลท์มากที่สุดที่ปิดอยู่ก็จะถูก Lock-Out ให้เปิดวงจรอยู่อย่างนั้น เมื่อตัว Circuit Breaker ปิดลงอีกครั้ง สวิตช์ตัดคอนทั้งหมดที่อยู่ด้านแหล่งจ่ายไฟซึ่งนับจากจุดที่เกิดฟอลท์ก็จะปิดลงตามลำดับดังภาพที่ 1

จากภาพที่ 1 ในระบบนี้ทางด้านแหล่งจ่ายจากจุดที่เกิดฟอลท์จะทำการจ่ายไฟฟ้าในระบบอีกครั้งหลังจากเกิดฟอลท์ อย่างไรก็ตามในการเกิดฟอลท์แต่ละครั้งนั้นจะทำให้เกิดผลกระทบในส่วนอื่น ๆ ด้วยนั่นคือ ด้านที่เป็นโหลดซึ่งการขัดข้องในครั้งนี้จะเกิดขึ้นไปเรื่อย ๆ จนกว่าจะมีการทำการซ่อมบำรุงแล้วเสร็จ ในการลดปัญหาที่เกิดขึ้นได้นั้นจะต้องอาศัยระบบควบคุมระยะไกล (Remote Terminal Unit ; RTU) ซึ่งจะใช้ร่วมกับระบบการสื่อสารโทรคมนาคม ซึ่งตัว Remote Control Unit และสายเคเบิลของระบบการสื่อสาร จะถูกติดตั้งในทุก ๆ ส่วนในระบบจำหน่ายที่เป็น Loop Switch เมื่อฟอลท์เกิดขึ้นที่ส่วนใด ส่วนที่ไม่เกิดฟอลท์ที่อยู่ในส่วนถัดไปในระบบจำหน่ายจะได้รับการจ่ายไฟฟ้าอีกครั้ง โดยการหน่วงเวลาของระบบ (Time delay system)

ภาพที่ 1

ภาพแสดง Isolation pocedure in a remote-control system



2.3 ระบบการส่งจ่ายอัตโนมัติโดยใช้การสื่อสาร

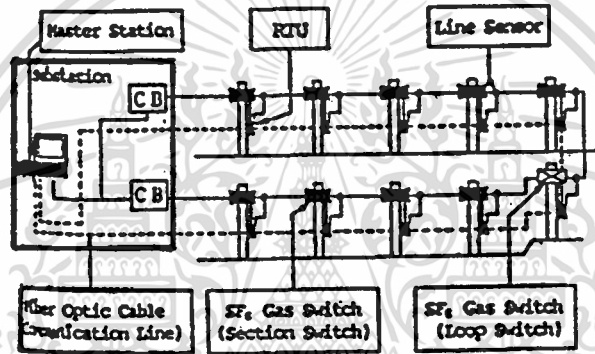
ระบบส่งจ่ายอัตโนมัติได้มีการพัฒนาระบบควบคุมระยะไกลโดยใช้สัญญาณวิทยุเข้าช่วยในพื้นที่ที่อยู่ห่างไกลมาก ๆ ซึ่งระบบนี้จะใช้ค่าใช้จ่ายในการติดตั้งน้อยกว่าการใช้สายเคเบิลใยแก้ว และเมื่อไม่นานมานี้ได้มีผู้คิดค้นวิธีการควบคุมระยะไกลที่เรียกว่า Non-interrupting system ซึ่งวิธีการนี้ได้ถูกคิดค้นจากการเกิดฟอลท์ในระบบที่ส่วนใหญ่จะพบฟอลท์ที่เกี่ยวข้องกับ Ground และการที่กระแสฟอลท์มีไม่มากพอที่จะทำให้สวิตช์ตัดคอนทำงานโดยเฉพาะอย่างยิ่งความต้านทานของ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ระบบ Ground ซึ่งการขัดข้องในส่วนที่ไม่ได้เกิดฟอลต์สามารถป้องกันได้โดยใช้สวิตช์ตัดคอน สำหรับการตัดกระแสฟอลต์ในระบบจำหน่าย ซึ่งองค์ประกอบ สำหรับระบบ Non-interrupting system นี้ประกอบด้วย ศูนย์ควบคุม (Master Station) , Remote Terminal Unit , Line Sensor , Automatic Reclosing Sectionlizer และ Fiber Optic Cable ดังภาพที่ 2

ภาพที่ 2

แสดงโครงสร้างของ Non-interrupting System



2.4.1 ศูนย์ควบคุม (Master Station) อุปกรณ์ที่ติดตั้งในศูนย์ควบคุมจะแบ่งออกเป็น 4 ส่วนหลัก ๆ คือ (ภาพที่ 3)

2.4.1.1 ส่วนของคอมพิวเตอร์หลัก (Host- Computer Subsystem) ซึ่งทำหน้าที่ประมวลผลและเก็บข้อมูลหลักของระบบที่ส่งมาจากสถานีระยะไกลก่อนแสดงผลออกทางหน้าจอคอมพิวเตอร์

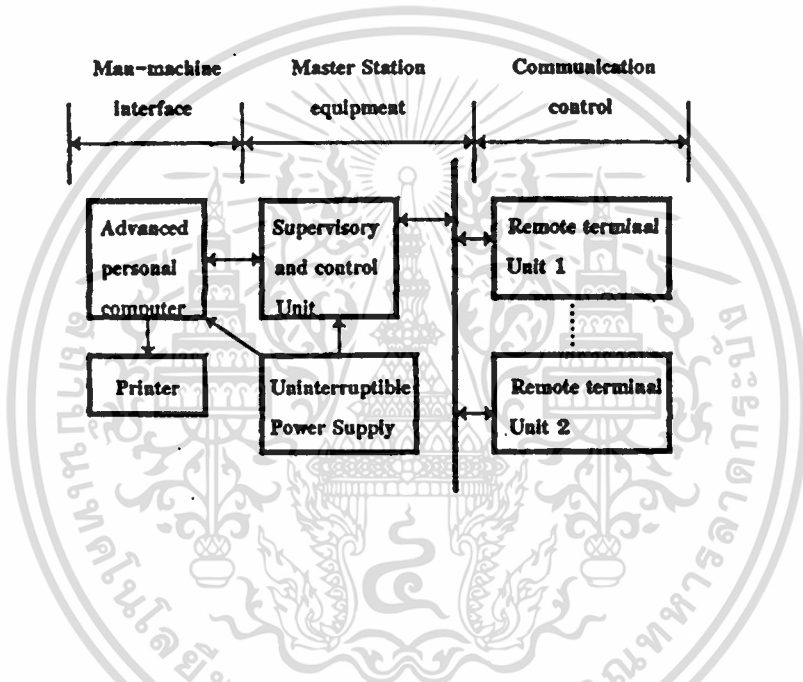
2.4.1.2 ส่วนของอุปกรณ์รับคำสั่งและแสดงผลข้อมูล (Man-Machine Interface Substation) ทำหน้าที่ในการสื่อความหมายระหว่างศูนย์กับคอมพิวเตอร์โดยแสดงผลในรูปของข้อความ สัญลักษณ์ทางจอภาพและเครื่องพิมพ์พร้อมทั้งมีสัญญาณเตือนภัย เพื่อแจ้งเหตุและถ่ายทอดคำสั่งของผู้ควบคุมผ่านเข้าไปทางเครื่องคอมพิวเตอร์ได้โดยอุปกรณ์ของแป้นพิมพ์(Keyboard)

2.4.1.3 ส่วนของอุปกรณ์ควบคุมการสื่อสารข้อมูล (Communication Controller Subsystem) ซึ่งทำหน้าที่ในการจัดระบบการสื่อสารข้อมูลระหว่างศูนย์ควบคุมกับสถานีระยะไกลที่เชื่อมโยงถึงกัน อุปกรณ์เหล่านี้เป็นหัวใจสำคัญในการตอบสนองข้อมูลที่รวดเร็วเพื่อรายงานความผิดปกติที่เกิดขึ้นให้ผู้ควบคุมทราบและทำการแก้ไขต่อไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.4.1.4 ส่วนของอุปกรณ์ระบบจ่ายไฟสำรองให้กับ 3 ส่วนหลักข้างต้นเมื่อเกิดไฟฟ้าดับหรือขัดข้อง

ภาพที่ 3
แสดงลักษณะการทำงานของ Master Station



2.4.2 ระบบสื่อสาร (Communication media) การส่งถ่ายที่มีความหมายจากแหล่งข้อมูลหนึ่งไปสู่อีกแหล่งข้อมูลหนึ่ง สามารถแบ่งเป็น 2 ประเภทคือ การสื่อสารตามสายและการสื่อสารทางวิทยุสื่อสาร จากทั้ง 2 ประเภทสามารถแบ่งเป็นส่วน ๆ ได้ดังนี้คือ (ภาพที่ 4)

2.4.2.1 แหล่งข้อมูลซึ่งผู้ควบคุมจะเป็นผู้จัดส่งและรับข้อมูลหรือส่งด้วยภาพ ฯลฯ

2.4.2.2 อุปกรณ์ส่งข้อมูล จะทำหน้าที่เปลี่ยนข้อมูลจากแหล่งข้อมูลในรหัสและส่งสัญญาณไปที่ช่องของการสื่อสารหรือช่องสัญญาณนั้น ๆ

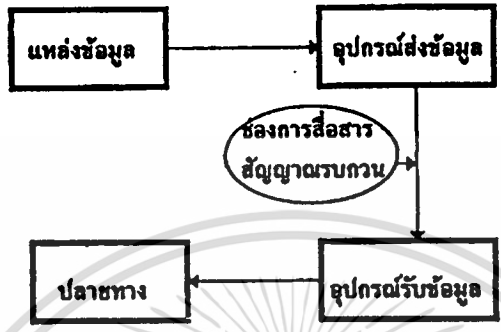
2.4.2.3 อุปกรณ์รับข้อมูล เมื่อมีการการรับสัญญาณจากเครื่องส่งกำลังเครื่องรับสัญญาณจะมีกำลังอ่อนลงและยังมีสัญญาณรบกวนมาแทรกซึ่งเครื่องรับจะต้องมีความสามารถเลือกรับและขยายเอาเฉพาะสัญญาณความถี่ที่ต้องการ

2.4.2.4 ปลายทาง เป็นอุปกรณ์ทางไฟฟ้าที่แปลงสัญญาณออกมาในลักษณะที่เป็น

ข่าวสาร

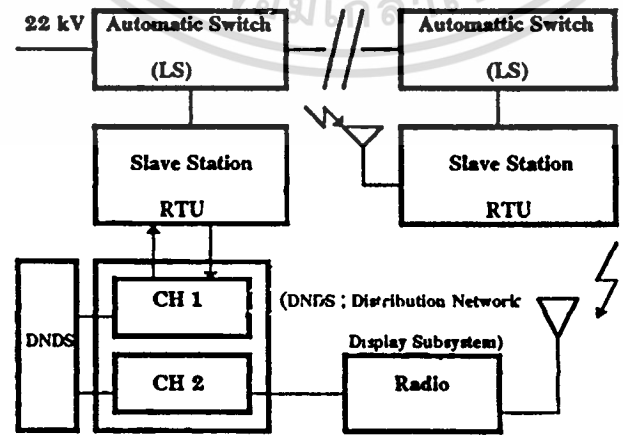
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาพที่ 4
แสดงส่วนต่าง ๆ ของระบบสื่อสารเบื้องต้น



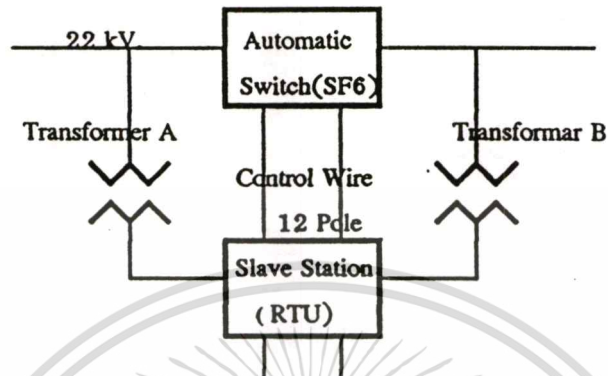
2.4.3 สถานีระยะไกล(Remote Terminal Unit ; RTU) เป็นอุปกรณ์ลูกข่ายที่ใช้ในการควบคุมการ เปิด-ปิด อุปกรณ์ตัดตอน(สวิตช์ตัดตอนแรงสูงที่ดับการอาร์คด้วยแก๊ส SF6) ที่ติดตั้งอยู่ในระบบจำหน่ายโดยรับคำสั่งควบคุมการทำงานจากแม่ข่ายที่ติดตั้งอยู่ที่ศูนย์ตั้งการ โดยแบ่งเป็น 2 ชนิดคือ ชนิดที่ควบคุมด้วยสายเคเบิล(Cable Type)และชนิดที่ควบคุมด้วยสัญญาณวิทยุสื่อสาร (Radio Type) ดังภาพที่ 5 และภาพที่ 6

ภาพที่ 5
แสดงลักษณะของ Remote Terminal Unit



ภาพที่ 6

แสดงส่วนประกอบของชุด Remote Terminal Unit



จากภาพที่ 6 เราสามารถสั่งการให้สวิตช์ทำงานได้จากทั้งศูนย์ควบคุมและจาก Local switch บน Remote terminal Unit แต่ละตัว ซึ่ง Remote terminal Unit ปกติแล้วจะรับกำลังไฟฟ้าจากหม้อแปลงแรงดันต่ำที่ทำหน้าที่ป้องกันการ Operation ในกรณีที่เกิดการตัดของระบบ Lock-Out นอกจากนี้ Remote terminal Unit ยังประกอบด้วยวงจรตรวจจับฟอลต์ซึ่งได้รับข้อมูลจาก Line Sener โดยแยกกันระหว่าง Short Circuit Fault และ Ground Fault ซึ่งแสดงลักษณะรูปร่างของ Remote terminal Unit ได้ดังภาพที่ 7 ซึ่งจะป็นรูปทรงกระบอกและมีกล่องเหล็กรูปทรงสี่เหลี่ยมผืนผ้าหุ้มอีกชั้นหนึ่งซึ่งภายในประกอบด้วยวงจรควบคุมสวิตช์ตัดตอนและวงจรสื่อสาร ซึ่งสาเหตุที่ต้องห่อหุ้มอีกชั้นเอาไว้ก็เพื่อป้องกันความชื้นและสัญญาณรบกวนอื่น ๆ

ภาพที่ 7

แสดงรูปร่างของ Remote terminal Unit



ภาพที่ 8
แสดงการติดตั้ง Line Sensor ในระบบจำหน่าย



และจากภาพที่ 8 เป็นอุปกรณ์ Sensor ที่มีหน้าที่วัดกระแสในแต่เฟสและส่งข้อมูล ไปให้ Remote terminal Unit โดยผ่านทาง Fiber Optic Cable และ Sensor เหล่านี้มีหลายขนาดให้เลือกเพื่อให้เข้ากับสายส่งในระบบจำหน่ายดังคุณลักษณะตามตารางที่ 1 และการติดตั้งสามารถทำได้ภายใต้สภาวะฮอตไลน์ (Hot Line Condition) ดังภาพที่ 8 และแสดง Block Diagram ได้ดังภาพที่ 9

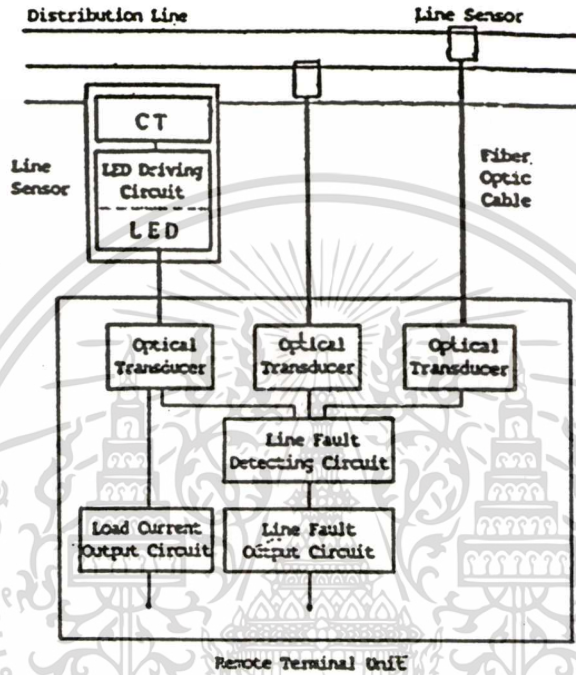
ตารางที่ 1

แสดง Characteristic ของ Line Sensor

Rated voltage	12 kV	24 kV
Rated current	400 A	
Impulse withstand voltage	95 kV	150 kV
Power frequency withstand voltage	350 kV	50 kV.
Rated short-time withstand voltage	13kV	
Setting of fault detection	Short-circuit fault	600 A
	Ground fault	10 A

ภาพที่ 9

แสดง Block Diagram ของ Line Sensor

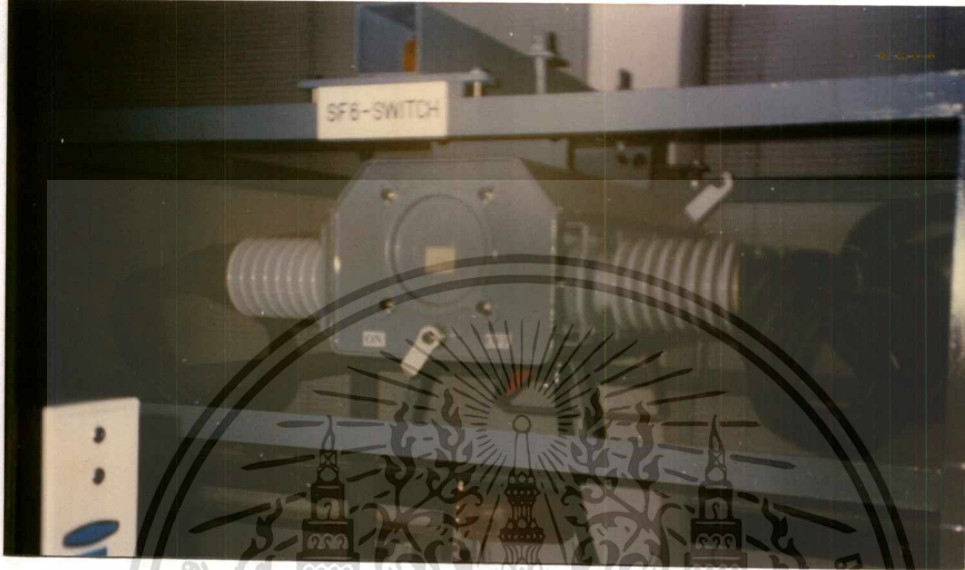


จากที่กล่าวมาแล้วข้างต้นว่า ระบบ Non-Interruption ได้มีการนำ Remote terminal Unit มาควบคุมการทำงานของ Automatic reclosing sectionlizer (SF6) ซึ่งเป็นสวิตช์ที่ใช้ก๊าซเป็นตัวดับอาร์ค ที่มีคุณภาพสูงซึ่งถูกหุ้มด้วยวัสดุประเภทเหล็กสแตนเลส ดังภาพที่ 10 ตัวสวิตช์นี้มีความทนทานสูงเพราะ ก๊าซ SF6 มีสภาพความเป็นฉนวนที่ดีและสวิตช์นี้ใช้หลักการอาร์คที่เกิดขึ้นในกระบอกลม(Puffer Cylonder System) จะถูกใช้เป็นกลไกในการทำงานดังภาพที่ 11 SF6 นี้จะบรรจุอยู่ในกระบอกลมซึ่งภายในมีหน้าสัมผัส (Contact) เมื่อวงจรเปิดออกก๊าซจะถูกเป่าเข้าไปใน Electrode และจะทำให้เกิดการทำงานขึ้นที่ระดับความดัน 1.5 bar และที่กระแสฟอลท์ตั้งแต่ 1.5 kA. ขึ้นไป ในระบบนี้อุปกรณ์ที่ใช้ในการ Lock-Out จะถูกนำมาใช้ ซึ่งจะทำการ Lock-Out ก็ต่อเมื่อความดันของก๊าซลดลง คุณลักษณะของ Automatic reclosing sectionlizer (SF6) พิจารณาได้ดังตารางที่ 2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

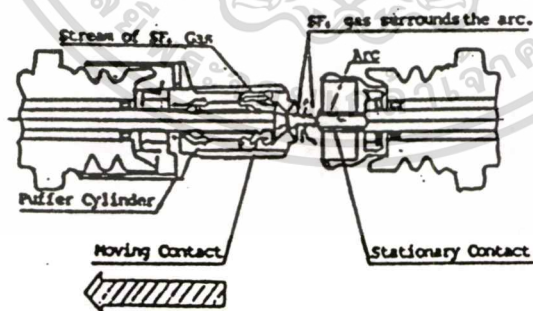
ภาพที่ 10

แสดงลักษณะรูปร่างของ Automatic reclosing sectionlizer (SF6)



ภาพที่ 11

แสดง Arc-extinguishing mechanism



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 2

แสดง Characteristic SF6 Gas

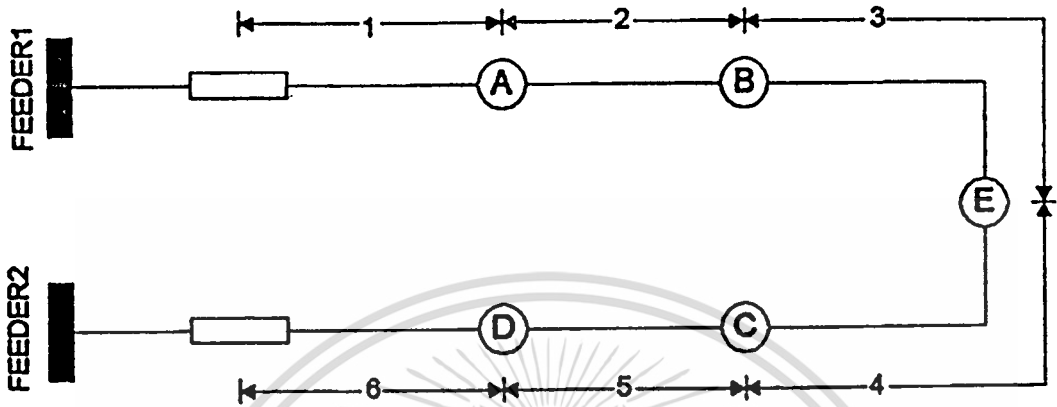
Rated voltage	12 kV	24 kV
Rated current	400 A	
Impulse withstand voltage	95 kV	150 kV
Power frequency withstand voltage	350 kV	50 kV
Rated short-circuit making current	32.5 kA	
Rated short-time withstand current	13 kA	
Breaking capacity of load current	400 A, 200 times	
Breaking capacity of ground fault current	1.5 kA, 100 times	
Mechanical endurance	500 times	

2.4 การทำงานของอุปกรณ์ป้องกัน

อุปกรณ์ป้องกันที่เรียกว่า Automatic reclosing sectionlizer เป็นอุปกรณ์ใหม่อย่างหนึ่งที่น่ามาติดตั้งในสายป้อนส่วนที่เป็นสายป้อนหลัก (main feeder) โดยทำหน้าที่เป็น switch สำหรับแยกสายป้อนส่วนที่ผิดปกติถาวร (permanent fault) ออกแล้วจ่ายไฟให้กับสายป้อนในส่วนที่เหลือให้สามารถจ่ายไฟได้โดยอัตโนมัติในระยะเวลาอันสั้น แต่เดิมเมื่อเกิด permanent fault ขึ้นในสายป้อนหลัก ไฟจะดับทั้งสายป้อน (feeder) เป็นระยะเวลานานจนกว่าจะแก้ไข fault ได้สำเร็จ ทำให้เกิดความเสียหายแก่ผู้ใช้ไฟฟ้าทั้งทางด้านเศรษฐกิจและสังคมเป็นอันมากการติดตั้งใน 1 ชุดจะประกอบด้วย sectionalizer 5 ตัว กับ 2 สายป้อน โดยแต่ละตัวจะแบ่งสายป้อนหลักออกเป็น ส่วน ๆ (section) ดังภาพที่ 12 จะเห็นว่าแต่ละชุดจะประกอบไปด้วย 10 ส่วน เมื่อเกิด fault ขึ้นที่ส่วนใด sectionalizer จะทำงานโดยอัตโนมัติ ให้ปลดส่วนนั้นออก ส่วนที่เหลือจะจ่ายไฟตามปกติ เช่น เกิด fault อยู่ในส่วนที่ 2 ในขั้นสุดท้าย ส่วนนี้จะถูกตัดออกโดยอัตโนมัติ ส่วนที่ 1 จะจ่ายไฟจาก สายป้อน 1 ส่วนที่ 3 จะจ่ายไฟจาก สายป้อน 2 (โดย sectionalizer (E)) จะเป็นตัวเชื่อมในการจ่ายไฟ ซึ่งในภาวะปกติแล้ว สายป้อน 1 และ 2 จะแยกกันโดยอิสระ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาพที่ 12
แสดงการติดตั้ง Automatic reclosing sectionlizer



2.4.1 ส่วนประกอบ Automatic Reclosing Sectionalizer จะเห็นว่า sectionalizer มีหน้าที่หลักในการตัดต่อส่วนที่เกิด fault เองโดยอัตโนมัติ (Aoki 1986 : 229-9) ดังนั้นส่วนประกอบใหญ่จะแบ่งได้เป็น 2 ลักษณะ คือ

2.4.1.1 switch เปิด-ปิด switch ที่ใช้เปิดหรือปิดจะต้องทนกระแสต่อเนื่อง (Continuous current) และสามารถเปิด-ปิด ขณะมีโหลด (load) ได้ สามารถสืบค้นหา fault ได้ แต่ไม่สามารถปลดขณะเกิด fault ได้ อุปกรณ์ที่ปลด fault ได้มีเพียง circuit breaker เท่านั้น ดังนั้นตัวนี้จะทำงานปลดเองขณะที่ไม่มีไฟเท่านั้น นอกจากนี้ส่วนประกอบภายนอกจะต้องมีน้ำหนักที่ค่อนข้างเบา การบำรุงรักษาบ่อย ติดตั้งภายนอกอาคาร (outdoor) ได้ซึ่ง switch ที่ใช้เป็นแบบสูญญากาศ (vacuum switch) คุณสมบัติต่าง ๆ เหมาะสมในการใช้งานแ่งงานวิจัยในวิทยานิพนธ์นี้จะใช้ Relay แบบ 4 contact มาเป็นตัว switch แทนโดยจุดหลักของ switch ที่กล่าวถึงนี้คือ ขดลวดภายในตัว Relay ซึ่งจะทำหน้าที่เป็นควบคุมการทำงานของหน้าสัมผัส (Contact) กล่าวคือ เมื่อมีกระแสไฟฟ้าไหลผ่านขดลวด ขดลวดก็จะเกิดสนามแม่เหล็กขึ้นและสนามแม่เหล็กจะมีแรงมากพอที่จะดึงหน้าสัมผัสให้เปลี่ยนสถานะการทำงาน และเมื่อไม่มีกระแสไฟฟ้าไหลผ่านขดลวดใน Relay ก็จะไม่เกิดสนามแม่เหล็ก ดังนั้น หน้าสัมผัสก็จะถูกสปริงดึงกลับให้กลับเป็นสถานะเดิม

2.4.1.2 วงจรควบคุม switch ดัชนีที่ใช้ในการควบคุมการทำงานของ switch ให้เปิดหรือปิดนั้นขึ้นอยู่กับแรงดัน (voltage) ส่วนจะให้ปิดหรือเปิดนั้นขึ้นอยู่กับชนิดของวงจรควบคุม switch ที่จะได้กล่าวต่อไป (K.Kato , H.Nagasaka , A.Okimoto , M.Akizuki 1986 :No.3.41) นอกจากนี้การทำงานของ sectionalizer แต่ละตัวจะต้องสัมพันธ์กันโดยใช้เวลา(time)เป็นตัวควบคุมช่วงเวลาต่างๆ เพื่อความสะดวกในการอธิบายการทำงานจะใช้อักษรต่าง ๆ แทน เช่น X,Y,Z เป็นต้น ซึ่งอักษรเหล่านี้ไม่ได้มีการกำหนดเป็นมาตรฐานไว้แต่อย่างใดเวลาที่ใช้ในการควบคุมประกอบด้วย

- Z-time คือเวลาที่ไ้หน่วง (delay) เพื่อไม่ให้ switch ทำงานทันทีที่รับสัญญาณ ดังภาพที่ 13 (a) และในส่วนของ Sectionalizer A,B,C,D ในขณะที่ switch ปิด (close) ใช้งานอยู่ เมื่อเกิดไฟฟ้าดับ switch จะยังคงอยู่ในสภาพปิดจนกว่าระยะเวลา Z-time จะหมดไป switch จึงจะถูกเปิดออกโดยอัตโนมัติ ในขณะที่ switch เปิดอยู่ (ใช้กับ sectionalizer (E) ที่เชื่อมสายป้อนเท่านั้น)เมื่อได้รับสัญญาณให้ปิดอันเนื่องจากสายป้อนใดสายป้อนหนึ่งที่เชื่อมต่ออยู่เกิดดับขึ้นสัญญาณนี้จะยังไม่สามารถรับรู้ได้โดยวงจรควบคุมสำหรับปิด เนื่องจาก Z-time จะเป็นตัวหน่วงเวลาเอาไว้เมื่อหมดช่วงเวลา Z-time สัญญาณนี้จึงจะไปควบคุมเวลาสำหรับปิด switch ให้ทำงาน

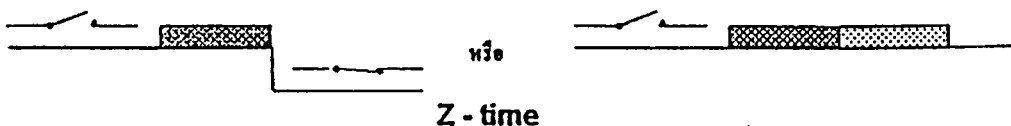
- X-time เป็นช่วงเวลาที่หน่วงไว้เพื่อไม่ต้องการให้ switch ปิดทันทีที่ได้รับสัญญาณ ไฟฟ้ามาใหม่ X-time จะเริ่มขึ้นในขณะที่ switch เปิดอยู่เสมอเมื่อสิ้นสุด X-time switch จะปิดดังภาพที่ 13 (b)

- XL-time จะเป็นช่วงเวลาสำหรับ sectionalizer เชื่อมระหว่างสายป้อนเท่านั้น (E) ในขณะที่ switch อยู่ในตำแหน่งเปิด XL-time จะเริ่มขึ้นในขณะที่ระยะเวลา Z-time หมดไป หลังจาก XL-time สิ้นสุดลง switch จะปิดทันทีดังภาพที่ 13 (c) ซึ่งความแตกต่างระหว่าง X-time กับ XL-time นั้นคือช่วงของเวลาและการรับสัญญาณ

-Y-time จะเป็นช่วงเวลาที่ใช้ตรวจสอบว่ามี fault เกิดขึ้นหรือไม่ ถ้ามีจะดีออก (lookout) ไม่ให้ switch สามารถทำงานโดยอัตโนมัติได้ต่อไป ถ้าในช่วงนี้ไม่มี fault เกิดขึ้นจะกลับคืนสู่สภาพปกติ (reset) ดังภาพที่ 13 (d) โดย Y-time จะเริ่มทันทีที่สิ้นสุด X-time ดังนั้น Y-time จะเกิดในขณะที่ switch ปิด เกิดขึ้นกับ sectionalizer ที่เชื่อมสายป้อนในขณะที่ XL-time ดำเนินไปอยู่ ณ มีสัญญาณให้ switch ไม่ต้องปิด เช่น การมีไฟกลับมา เนื่องจากการ reclose ของ circuit breaker และ sectionalizer Y-time จะเริ่มขึ้นทันทีที่ดังนั้น Y-time จะเริ่มขึ้นในขณะที่ switch เปิดอยู่

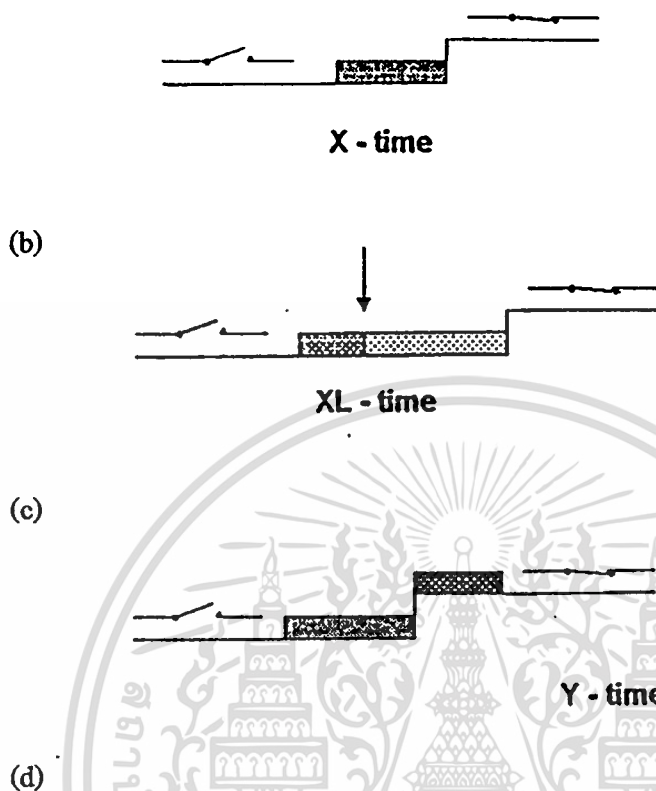
ภาพที่ 13

แสดงเวลาที่ไ้ใช้ในการทำงานของ Sectionalizer



(a)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



2.4.2 ชนิดของวงจรควบคุม จากที่กล่าวมาแล้วว่าการติดตั้งระบบป้องกันใน 1 ชุด ประกอบไปด้วย sectionalizer 5 ตัวในการควบคุม switch ให้เปิด-ปิด ซึ่งแบ่งออกได้เป็น 3 ชนิด คือ

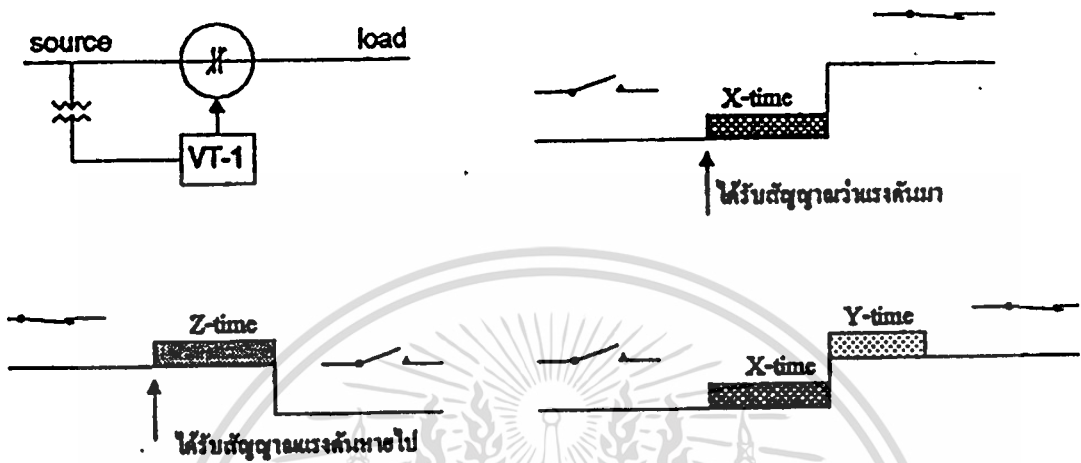
2.4.2.1 Voltage-time Control (VT-1 type) หรือเรียกว่า Fault Detecting Relay (FDR) ซึ่งจะรับสัญญาณแรงดันในการควบคุม switch โดยอาศัยหม้อแปลง (potential transformer) จากทางด้าน source ด้านเดียว ในสภาวะปกติ (มีไฟ) switch ปิดอยู่ดังภาพที่ 14 พิจารณาจากภาพที่ 14 การเปิด switch (Open) เมื่อแรงดันจากทางด้าน Source หายไป VT-1 จะเริ่ม Z-time ในขณะที่ switch ยังคงปิดอยู่และเปิดเมื่อครบ Z-time และเมื่อพิจารณาการปิด switch (Close) ในขณะที่ switch เปิดอยู่ VT-1 รับสัญญาณว่ามีแรงดันมาจะเริ่ม X-time และเมื่อสิ้นสุด X-time switch จะปิดและถ้า VT-1 lockout Y-time จะเริ่มขึ้นทันทีที่ switch จะปิดหลังจาก X-time สิ้นสุด ถ้าพบว่าไฟหายในช่วงนี้ VT-1 จะล๊อค (lockout) ไม่ให้วงจรควบคุมทำงานหลังจาก switch เปิดแล้ว และเมื่อ VT-1 Reset นั่นคือ การกลับคืนสู่สภาพเดิมของวงจรซึ่งเป็นไปได้ 2 ลักษณะคือ

ก) Reset โดยอัตโนมัติ ซึ่งจะเกิดขึ้น เมื่อหมดระยะเวลา Y-time แล้ว โดยไม่มีไฟดับวงจร VT-1 จะกลับคืนสู่สภาพเดิม (Reset) ได้เอง

ข) Reset ด้วยมือ VT-1 จะต้อง Reset ด้วยมือเท่านั้น หากเกิดการ lockout ขึ้น Sectionalizer แบบ VT-1 นี้จะติดตั้งที่ตัวแรก จาก Source นั่นคือ A และ D

ภาพที่ 14

แสดงการทำงานของวงจรควบคุม Sectionalizer แบบ VT-1



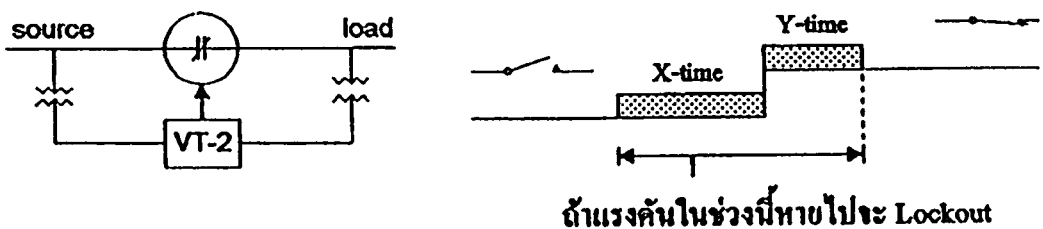
3.1.2.2 แบบ VT-2 (Voltage time control) VT-2 ได้รับสัญญาณในการควบคุมจากหม้อแปลงทางด้าน Source หรือ load อย่างใดอย่างหนึ่งเท่านั้น ในสภาวะปกติ (มีไฟ) Switch ปิด อยู่ทิศทางการจ่ายไฟคือจาก Source ไปยัง load ส่วนการเปิด switch (Open) เหมือน VT-1 และการปิด switch (Close) ก็เหมือนกันกับ VT-1 เช่นกันและถ้า VT-2 lockout จะ lock ไม่ให้ switch ทำงานใน 2 กรณีคือ

ก) เมื่อไฟดับระหว่าง X-time (ขณะที่กำลังรอเวลาที่จะสับเข้า) switch จะถูก lock ให้อยู่ในตำแหน่งปลด (open) ดังภาพที่ 15

ข) เมื่อไฟดับระหว่าง Y-time (หลังจาก switch สับเข้าแล้ว) ตัว switch จะถูกปลดออกและ lock ไม่ให้สับเข้าอีก และวงจรควบคุมจะไม่ทำงาน จนกว่าจะมีการ Reset ใหม่ด้วยมือดังภาพที่ 15

ภาพที่ 15

แสดงการทำงานของวงจร Sectionalizer แบบ VT-2



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดตทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

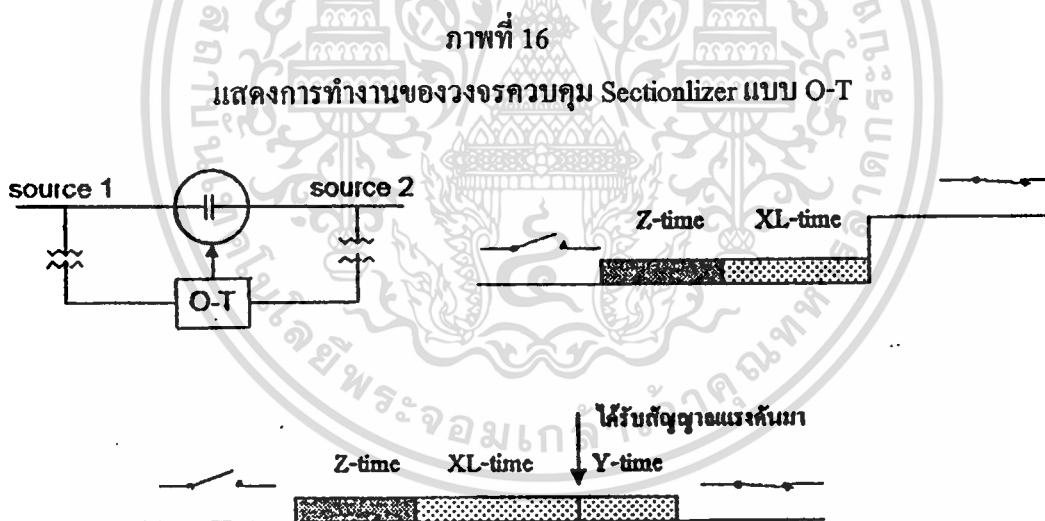
2.4.2.3 แบบ Open-tie (O-T) จะรับสัญญาณแรงดันในการควบคุม switch จาก Source 1 และ Source 2 ทั้งสองด้าน (2 สายป้อน) ในสภาวะปกติจะมีไฟอยู่ที่ 2 Source และ switch เปิดอยู่ การปิด switch (Close) เมื่อแรงดันจาก Source 1 หรือ Source 2 หายไปด้านใดด้านหนึ่ง XL-time จะเริ่มทำงานหลังจาก Z-time สิ้นสุดลง และ switch จะปิดเมื่อสิ้นสุด XL-time หลังจาก switch ปิดแล้วการควบคุมของ O-T จะถูกเปลี่ยนเป็น VT ทันทีในส่วนของ O-T Lockout ขณะที่อยู่ในช่วง XL-time นั้น O-T จะได้รับสัญญาณว่ามีแรงดันมาหยุด XL-time และเริ่ม Y-time ทันที ถ้าในช่วง Y-time แรงดันหายไป (พบ fault) O-T จะ lock ไม่ให้ switch ปิดได้และถ้า O-T Reset สามารถแยกออกเป็น 2 ลักษณะ คือ

ก) ถ้า switch lockout ในตำแหน่งเปิด O-T จะ Reset โดยอัตโนมัติ

ข) ถ้า switch lockout ในตำแหน่งเปิด (เปลี่ยนเป็นแบบ VT แล้ว) O-T จะ

Reset ด้วยมือเท่านั้น

Sectionalizer แบบ O-T นี้จะเป็นตัวเชื่อมระหว่าง 2 สายป้อนเข้าด้วยกัน ดังนั้นจะติดตั้งอยู่ที่กึ่งกลางระหว่าง 2 สายป้อน (E) ดังภาพที่ 16

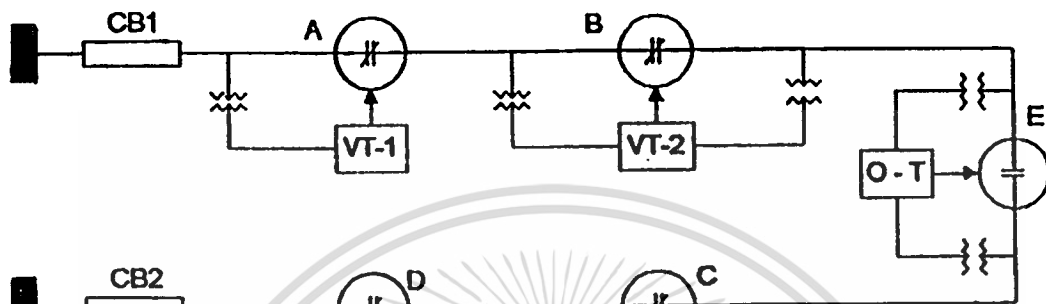


2.5 การทำงานของ Automatic Reclosing Sectionalizer

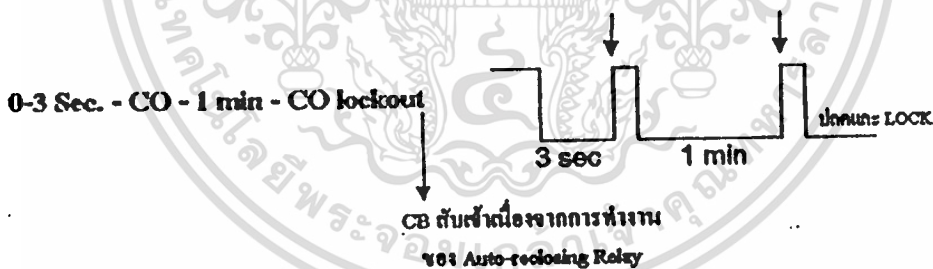
วงจรควบคุมการทำงานของ switch จะเป็นตัวกำหนดให้ switch เปิดหรือปิดเมื่อใดนั้นเป็นไปดังที่กล่าวไว้แล้ว นอกจากนี้การทำงานของวงจรควบคุมจะต้องสัมพันธ์กับการทำงานของ circuit breaker (CB) ที่สถานีย่อย Automatic reclosing Sectionalizer จะรับทราบการปลด-สับ ของ Circuit Breaker ก็โดยการรับสัญญาณว่ามีแรงดันหรือไม่มีของวงจรควบคุมการทำงานผ่านหม้อแปลงวัดแรงดัน ไฟฟ้าดังภาพที่ 17

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาพที่ 17
แสดงการติดตั้ง Automatic reclosing sectionlizerที่สมบูรณ์



ภาพที่ 18
แสดงการทำงานของ Circuit Breaker ที่กำหนด



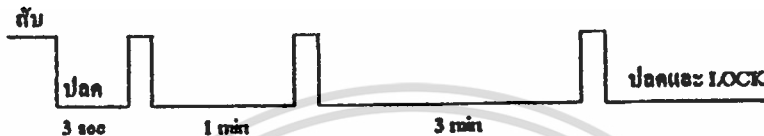
จากภาพที่ 18 แสดงให้เห็นว่า Automatic Reclosing Relay (Relay ที่ทำหน้าที่สั่งให้ circuit breaker สับใหม่ตามกำหนดเวลา) สั่งให้ Circuit Breaker สับใหม่ (reclose) ได้ 2 ครั้ง ครั้งแรก Circuit Breaker สับหลังจาก 3 วินาที เพื่อต้องการให้มีไฟจ่ายเร็วที่สุดในกรณีเกิด temporary fault เราจึงไม่ต้องการให้ Automatic reclosing sectionalizer ทำงานในกรณีที่เกิดฟอลท์ชนิดนี้ ดังนั้น sectionalizer จะรับรู้สัญญาณปลดของ Circuit Breaker ในช่วงปลด 1 นาที และรับสัญญาณการสับใหม่ของ Circuit Breaker หลังช่วงปลด 1 นาทีเพียงครั้งเดียว แต่การทำงานของ Automatic reclosing sectionalizer ต้องการรับสัญญาณการสับใหม่ของ Circuit Breaker 2 ครั้งคือ ครั้งแรกเพื่อตรวจสอบส่วนที่เกิดฟอลท์และครั้งหลังเพื่อจ่ายไฟตามเดิมในส่วนที่ไม่มีฟอลท์เราจำเป็นต้องเพิ่มการสับใหม่ของ Circuit Breaker อีกครั้งหนึ่งเป็นดังภาพที่ 19

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาพที่ 19

แสดงลำดับขั้นการทำงานของ Circuit Breaker ที่เพิ่มขึ้นเมื่อใช้กับ Automatic reclosing sectionlizer

0-3 Sec. - CO - 1 min - CO - 3 min - CO lockout



การทำงานของ Automatic reclosing sectionlizer สามารถแบ่งตามลักษณะของการเกิดฟอลต์ออกได้เป็น 2 กรณีคือ

2.5.1 เมื่อเกิด temporary fault โดยทั่วไปประมาณ 80% ฟอลต์ที่เกิดขึ้นส่วนใหญ่ในระบบจะเป็น temporary fault ซึ่งเมื่อเกิดขึ้นแล้วจะหายไป (clear) ภายในระยะเวลาอันรวดเร็ว เมื่อ Circuit Breaker ดับเข้าครั้งแรกก็สามารถจ่ายไฟได้ตามปกติ ในช่วงนี้จะต้องไม่ให้ Automatic reclosing sectionlizer ทำงานเพราะถ้าทำงานทันทีที่ Circuit Breaker ปลดออกจะทำให้ช่วงเวลาที่ไฟดับนานขึ้น เนื่องจากเมื่อ Automatic reclosing sectionlizer เปิดแล้วจะรอเวลาสำหรับปิด (X-time) หลังจากได้รับสัญญาณ ในภาพที่ 19 แสดงการทำงานของ sectionlizer A,B (ภาพที่ 15) กรณีไม่มี Z-time สมมุติเกิดฟอลต์ในช่วง BE สายป้อนช่วงนี้จะจ่ายไฟได้ตามปกติแต่ต้องรอให้ B ปิดเสียก่อน โดยใช้เวลา 3 sec. + X-time ของ A และ B ดังภาพที่ 20 ซึ่ง Z-time ผลที่ได้จึงไม่แตกต่างกับระบบเดิมที่การไฟฟ้าใช้อยู่ในปัจจุบัน

2.5.2 เมื่อเกิด Permanent fault เดิมเมื่อเกิด permanent fault ขึ้น Circuit Breaker ที่สถานีย่อยจะ lockout ทำให้ไฟดับทั้งสายป้อน Automatic reclosing sectionalizer จะช่วยแยกส่วนที่เกิด fault เองโดยอัตโนมัติแล้วดับไฟให้กับส่วนอื่นที่ไม่มี fault ตามเดิม Automatic reclosing sectionalizer ที่นำมาติดตั้งจะแบ่ง 2 สายป้อนออกเป็น 6 ส่วน (ภาพที่ 16) การทำงานในส่วน (section) ที่ 1,2 และ 3 จะเหมือนกับส่วนที่ 6,5 และ 4 ตามลำดับ ซึ่งแยกออกได้เป็น 3 กรณี

กรณีที่ 1 fault เกิดขึ้นในส่วนที่ 1 ดังภาพที่ 21 ในสภาพปกติ Circuit Breaker A และ B จะปิดและ E เปิดซึ่งมีขั้นตอนการทำงานดังนี้

ที่เวลา 0 วินาที

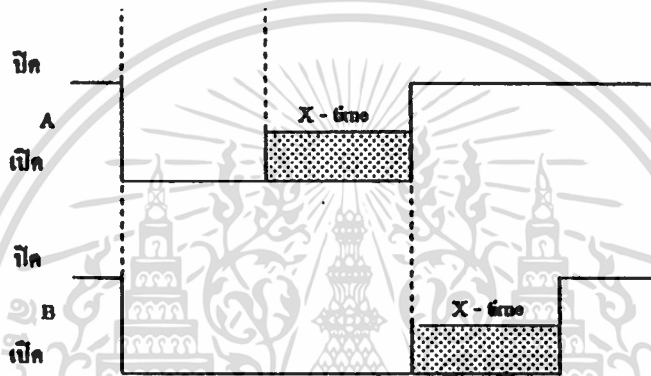
1) เกิดฟอลต์ที่จุด 1 Circuit Breaker 1 ปลดออก A,B และ E เริ่ม Z-time

ที่เวลา 3 วินาที

- 2) Circuit Breaker 1 ลับใหม่พบ fault ที่จุด 1 อีก จึงปลดออกช่วงที่ CB1
- 3) A และ B เปิด เนื่องจากครบ Z-time E เริ่ม XL-time เมื่อครบ Z-time

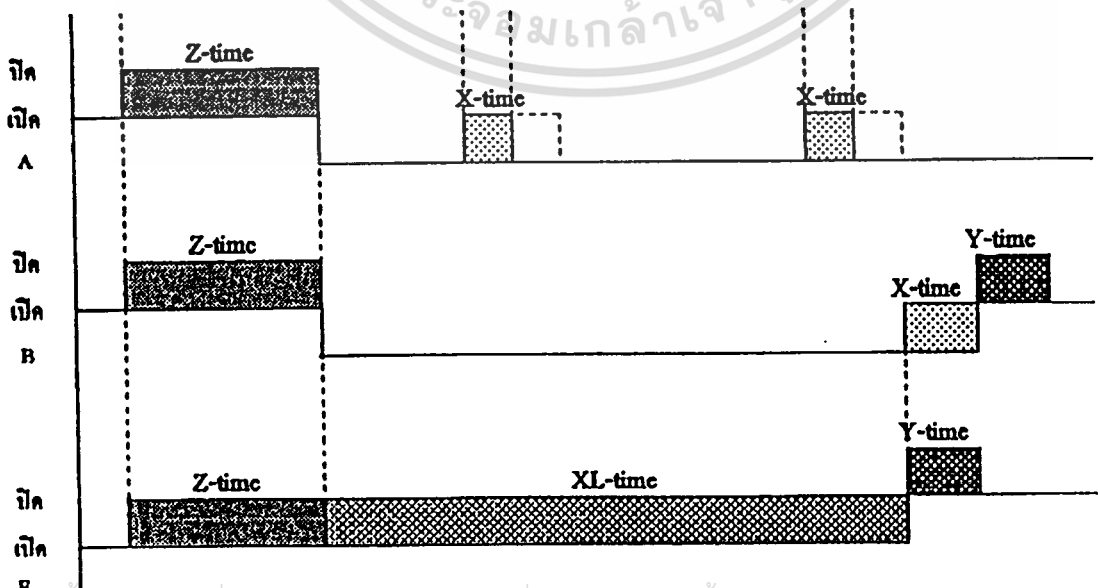
ภาพที่ 20

แสดงการทำงานของ Automatic reclosing sectionlizer A,B กรณีไม่มี Z-time



ภาพที่ 21

แสดงการทำงานของ Automatic reclosing sectionalizer เมื่อเกิดฟอลท์ในส่วนที่ 1



ที่เวลา 1 นาที

- 4) Circuit Breaker 1 สับใหม่ A เริ่ม X-time
- 5) Circuit Breaker 1 ปลดออกเพราะพบฟอลต์ที่ 1 อีกดังภาพที่ 22(a)ทำให้ A reset ตู้สภาพเดิม

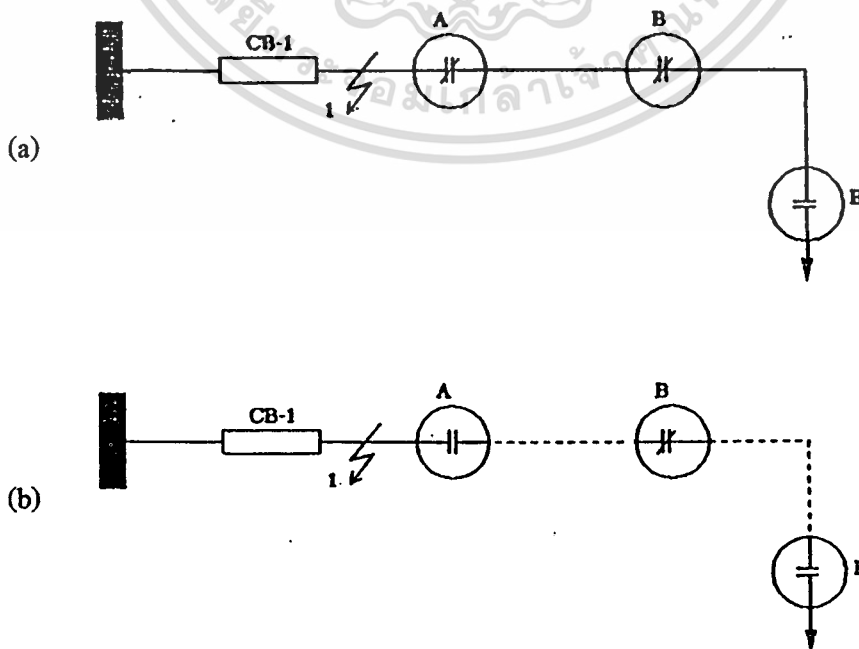
ที่เวลา 3 นาที

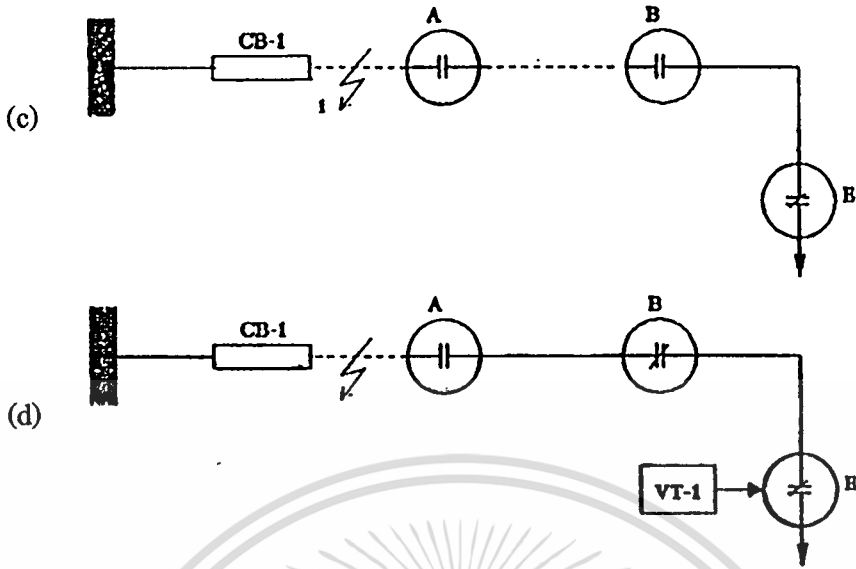
- 6) Circuit Breaker 1 สับใหม่ A เริ่ม X-time
- 7) Circuit Breaker 1 ปลดออก และ lockout ดังภาพที่ 22 (b)และทำให้ A reset ตู้สภาพเดิม
- 8) E ปิดเมื่อครบ XL-time ดังภาพที่ 22 (c) และ B เริ่ม X-time
- 9) B ปิดเมื่อครบ X-time และเริ่ม Y-time
- 10) B reset เมื่อครบ Y-time
- 11) ส่วนที่ 1 จะ ไม่มีไฟเท่านั้น ดังภาพที่ 22 (d)
- 12) E จะเปลี่ยนจาก OT เป็นแบบ VT-1

ภาพที่ 22

แสดงการทำงานของ Automatic reclosing sectionalizer

เมื่อเกิดฟอลต์ในส่วนที่ 1 ที่เวลาต่าง ๆ

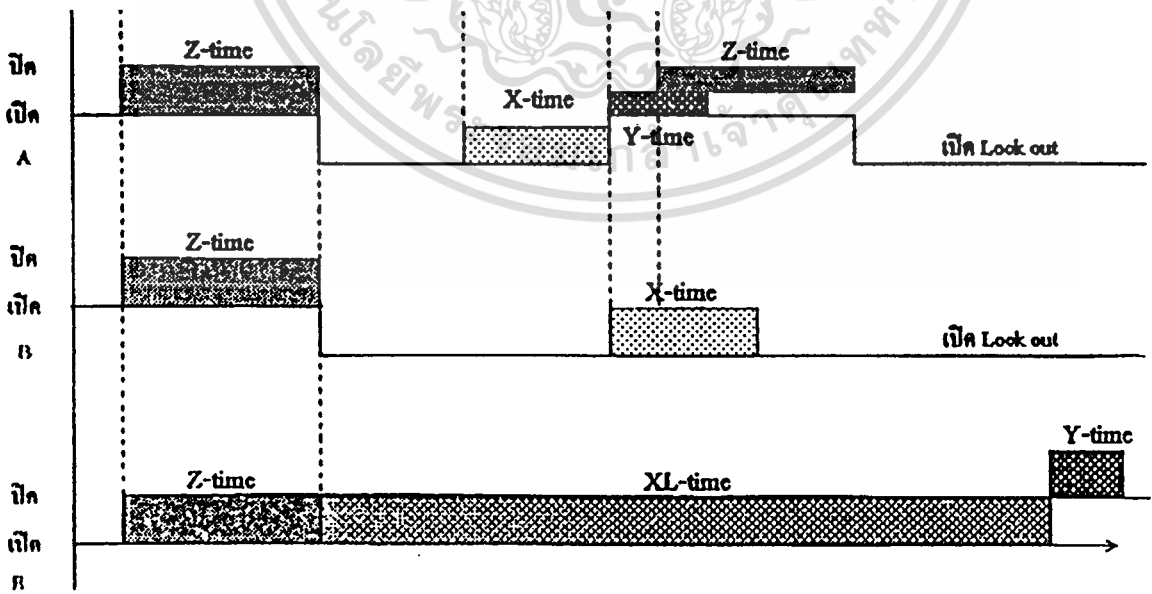




กรณีนี้ 2 fault เกิดขึ้นในส่วนที่ 2 ดังภาพที่ 23 ในสภาพปกติ Circuit Breaker 1 A และ B ปิด E อยู่ในสภาวะเปิดซึ่งมีขั้นตอนการทำงานดังนี้

ภาพที่ 23

แสดงการทำงานของ Automatic reclosing sectionalizer เมื่อเกิด fault ในส่วนที่ 2



ที่เวลา 0 วินาที

- 1) เกิด fault ที่จุด 2 Circuit Breaker 1 ปลด A,B ออกและ E เริ่ม Z-time

ที่เวลา 3 วินาที

- 2) Circuit Breaker 1 สับใหม่ พบ fault ที่จุด 2 อีก
- 3) A และ B เปิดเนื่องจากครบ Z-time E เริ่ม XL-time เมื่อครบ Z-time ดังภาพที่ 24 (a)

ที่เวลา 1 นาที

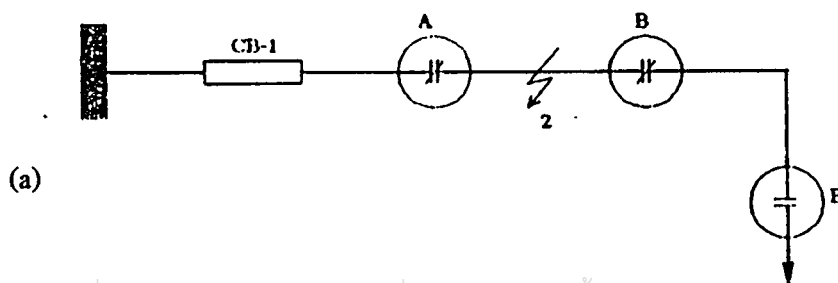
- 4) Circuit Breaker 1 สับใหม่ , A เริ่ม X-time
- 5) A จะปิดเมื่อครบ X-time , และเริ่ม Y-time
- 6) B เริ่ม X-time
- 7) Circuit Breaker 1 ปลดออก เพราะพบ fault ที่จุด 2 ดังภาพที่ 24 (b)
- 8) A เริ่ม Z-time จะเตรียมที่จะ lockout เพราะไฟหายในช่วง Y-time และ B จะ lockout โดยอยู่ในตำแหน่งเปิดเหมือนเดิม เพราะไฟหายในช่วง X-time ตามคุณสมบัติของ VT-2 ดังภาพที่ 24 (c)
- 9) เมื่อครบ Z-time A จะเปิดแล้ว lockout ดังภาพที่ 24 (d)

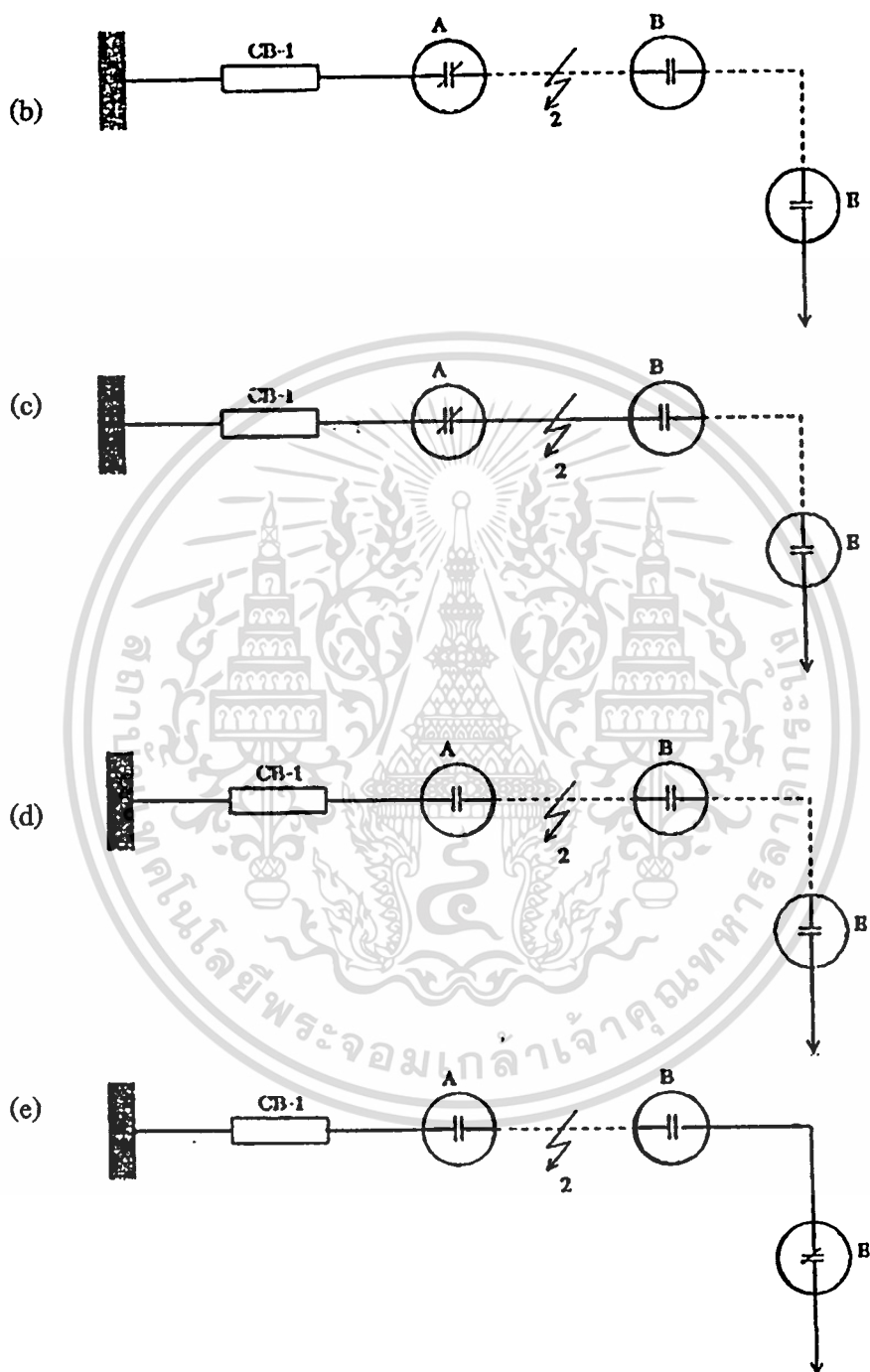
ที่เวลา 3 นาที

- 10) Circuit Breaker 1 สับใหม่ใช้งานได้
- 11) E จะปิดเมื่อครบ XL-time
- 12) ส่วนที่ 2 จะไม่มีไฟเท่านั้นดังภาพที่ 24 (e)
- 13) E จะเปลี่ยนจาก O-T เป็นแบบ VT-1

ภาพที่ 24

แสดงขั้นตอนการทำงานของ Automatic reclosing sectionalizer
ในช่วงเวลาต่าง ๆ



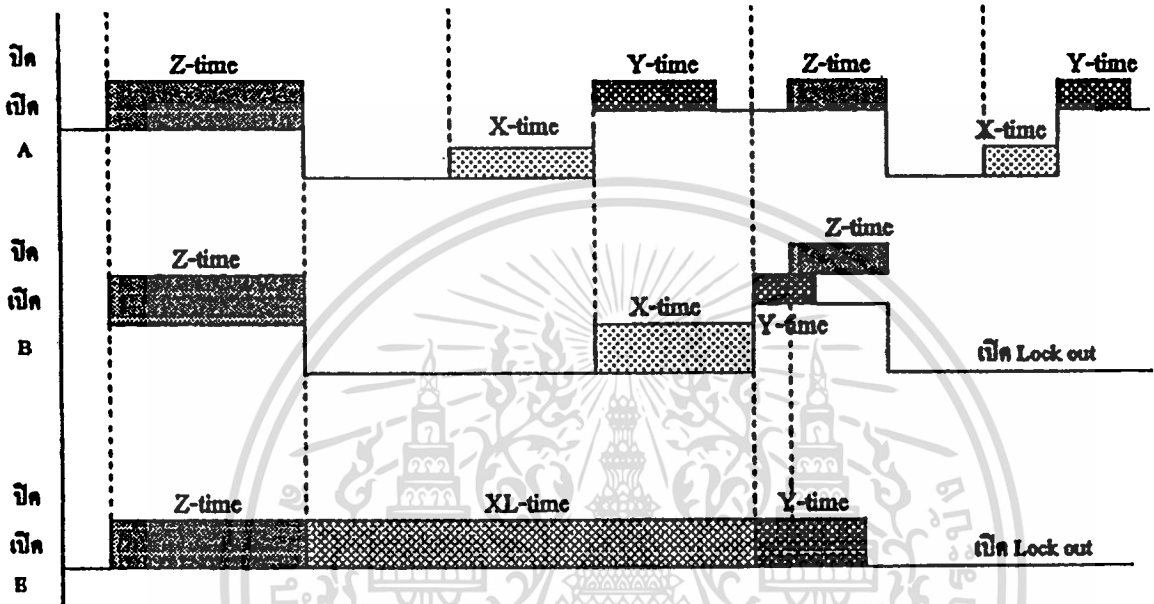


กรณีที่ 3 fault เกิดขึ้นในส่วนที่ 3 ดังภาพที่ 25 ในสภาพปกติ Circuit Breaker 1 A และ B ปิด E จะอยู่ในสภาวะเปิด ซึ่งมีขั้นตอนการทำงานดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาพที่ 25

แสดงการทำงานของ Automatic reclosing sectionalizer เมื่อเกิดฟอลท์ในส่วนที่ 3



ที่เวลา 0 วินาที

- 1) เกิด fault ที่จุด 3 Circuit Breaker 1 ปลดออก A,B และ E เริ่ม Z-time

ที่เวลา 3 วินาที

- 2) Circuit Breaker 1 สับใหม่ พบ fault ที่จุด 3 อีก จึงปลดออกนาน 1 นาที

ดังภาพที่ 26 (a)

- 3) A และ B เปิดเนื่องจากครบ Z-time ส่วน E เริ่ม XL-time เมื่อครบ Z-time

ที่เวลา 1 นาที

- 4) Circuit Breaker 1 สับใหม่ A เริ่ม X-time
- 5) A ปิดเมื่อครบ X-time และเริ่ม Y-time และ B เริ่ม X-time ดังภาพที่ 26 (b)
- 6) A reset เมื่อครบ Y-time
- 7) B ปิดเมื่อครบ X-time และเริ่ม Y-time ดังภาพที่ 26 (c)

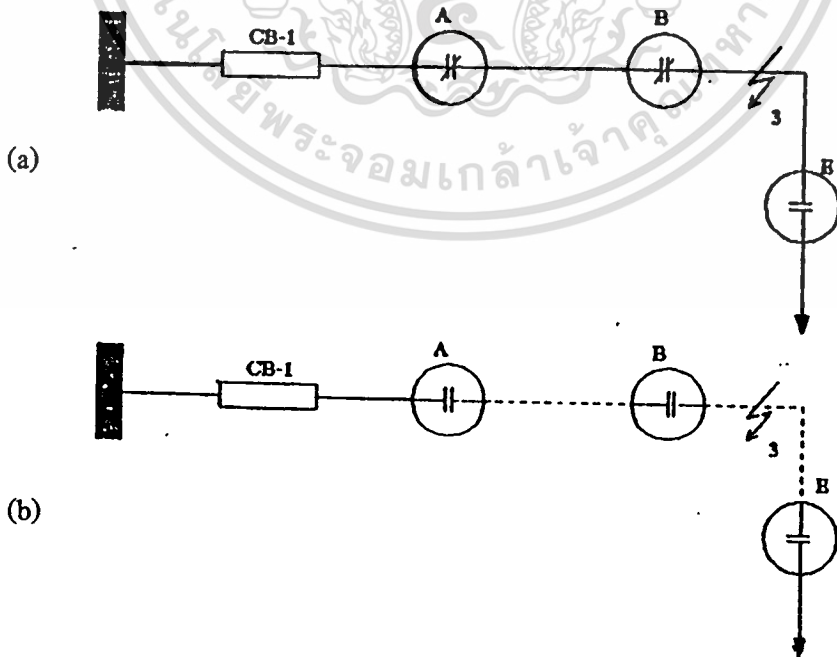
- 8) E เริ่ม Y-time เพราะพบว่ามีไฟมา
- 9) Circuit Breaker 1 ปลดออกเพราะพบ fault ที่ 3 อีก ดังภาพที่ 26 (d)
- 10) E เปิด (เหมือนเดิม) และ lockout เพราะไฟหายไปในช่วง Y-time
- 11) A เปิดเมื่อครบ Z-time B เปิดและ lockout เพราะไฟหายไปในช่วง Y-time
ดังภาพที่ 26 (e)

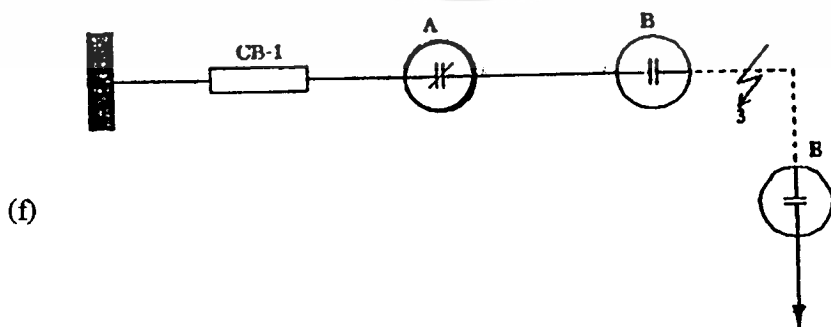
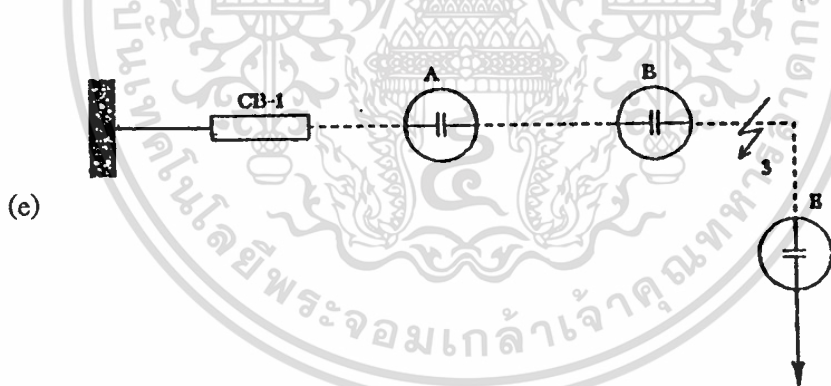
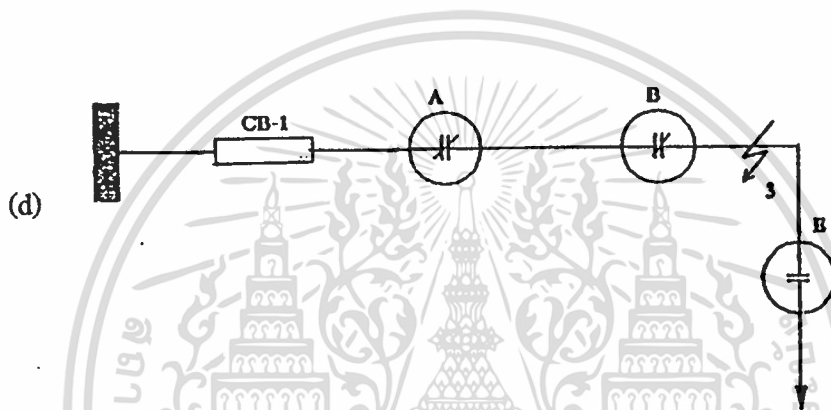
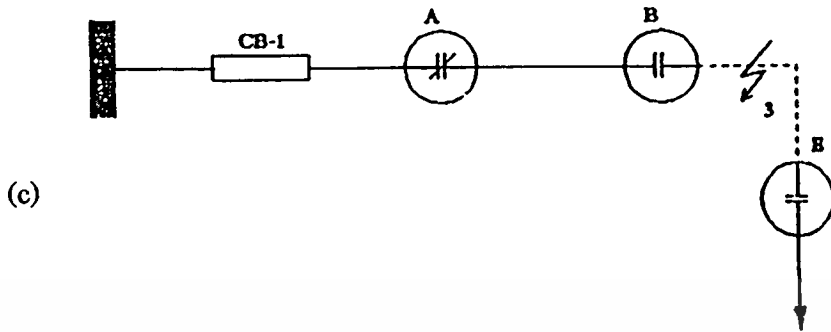
ที่เวลา 3 นาที

- 12) Circuit Breaker 1 ถัดใหม่ ใช้งานได้, A เริ่ม X-time
- 13) A ปิดเมื่อครบ X-time และเริ่ม Y-time
- 14) A reset เมื่อครบ Y-time
- 15) ส่วนที่ 3 จะ ไม่มีไฟเท่านั้น ดังภาพที่ 26 (f)

ภาพที่ 26

แสดงขั้นตอนการทำงานของ Automatic reclosing sectionalizer
เมื่อเกิดฟอลท์ในส่วนที่ 3 ช่วงเวลาต่าง ๆ





เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.6 เหตุผลในการเลือกวงจรควบคุมแต่ละชนิด

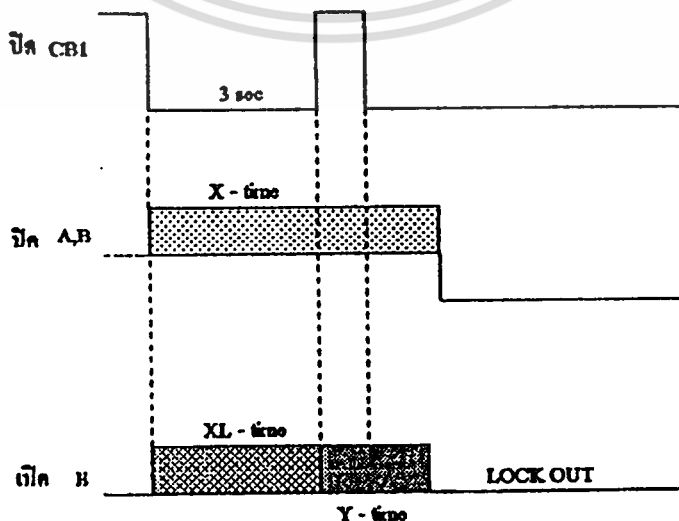
VT-1 รับสัญญาณจากด้าน source ด้านเดียว เนื่องจากไม่ต้องการให้ switch ปิดเมื่อได้รับสัญญาณทางด้าน load ดังในกรณีที่ 1 ถ้าปิดแล้ว A จะปิดไปพบ fault ทำให้อีกสายป้อนหนึ่ง (Circuit Breaker 2) ไฟดับไปด้วย

VT-2 รับสัญญาณจากด้านใดด้านหนึ่งของทั้ง source และ load เนื่องจากต้องการให้ switch ปิดจ่ายไฟจากอีก feeder หนึ่ง (Circuit Breaker 2) มาใช้แทน feeder เดิม (Circuit Breaker 1) ในกรณีที่ 1 และ 2 นอกจากนี้ VT-2 ต้อง lockout ในช่วง X-time ได้ เพราะไม่ต้องการให้ไฟจากอีกสายป้อนหนึ่ง (Circuit Breaker 2) ดับไปด้วย เมื่อ B สับไปพบ fault (ในกรณีที่ 2 ถ้า VT-2 ไม่ lockout ในช่วง X-time) O-T จะเปลี่ยนเป็นแบบ VT-1 เมื่อเชื่อมสายป้อนกันแล้ว เพราะหลังจากที่ switch ของ O-T ปิดแล้วจะไม่มี การเชื่อมสายป้อนกันอีกจะมีเพียงการ lockout เมื่อเกิด fault เท่านั้น O-T จะต้องมี Z-time ด้วย แม้ว่าจะอยู่ในสภาพเปิดอยู่ทันทีที่ได้สัญญาณว่าไฟข้างใดข้างหนึ่งหายไปเพื่อป้องกันการทำงานผิดพลาด เช่น ในกรณีที่ 1, 2 หรือ 3 ถ้า E ไม่มี Z-time มันจะเริ่ม XL-time ทันทีที่ CB1 ปลดออกและเริ่ม Y-time เมื่อ CB1 สับเข้าหลังจาก 3 วินาทีแล้ว ปรากฏว่า fault ยังอยู่ CB1 จะปลดออกอีกทำให้ E จับได้ว่าไฟหายไปในช่วง Y-time จึง lockout ดังภาพที่ 27

ภาพที่ 27

แสดงการทำงานของ Automatic reclosing sectionalizer

ในกรณีที่ E (O-T) ไม่มี Z-time



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากการทำงานของ sectionalizer ทั้ง 3 กรณีจะเห็นได้ว่า มันจะทำงานตามเวลาต่าง ๆ ที่ตั้งไว้ เมื่อได้รับสัญญาณแรงดันจากการปลดสับของ circuit breaker ที่สถานีย่อยเวลาจึงเป็นปัจจัยสำคัญที่จะให้ sectionalizer ทำงานผิดพลาดได้ถ้าเวลาไม่สัมพันธ์กัน ผู้ปฏิบัติงานจะต้องแน่ใจว่าเวลาที่ตั้งไว้ตามที่กำหนด (ไม่ได้กล่าวถึงในที่นี้) เสียก่อนที่จะนำไปใช้งานสำหรับการเกิดไฟดับในกรณีอื่น ๆ นอกเหนือจาก permanent fault เช่น under frequency , การดับไฟเพื่อปฏิบัติงาน , ผู้ปฏิบัติงานในส่วนที่ sectionalizer ติดตั้งอยู่จะต้องศึกษารายละเอียดจาก instruction manual ให้เข้าใจเสียก่อน และจะต้องเพิ่มความระมัดระวังเป็นพิเศษสำหรับการใช้งาน sectionalizer เพราะการทำงานโดยอัตโนมัติ หากรู้ทำไม่ถึงการณ์อาจเกิดความผิดพลาดได้เสมอ

2.7 การแยกฟอลต์ออกจากระบบจำหน่าย

จากที่กล่าวมาแล้วข้างต้น ระบบสื่อสารและ Line Senser มีความสำคัญมากที่จะทำให้สามารถตรวจสอบหาจุดที่เกิดฟอลต์ได้ทันทีหลังจากการเกิดฟอลต์ขึ้น ด้วยเหตุนี้เมื่อเกิดฟอลต์ขึ้นและตรวจจับฟอลต์ได้แล้วจึงจำเป็นต้องมาแยกประเภทการเกิดฟอลต์ในระบบจำหน่าย ซึ่งฟอลต์ในระบบจำหน่ายนั้นสามารถแยกเป็นฟอลต์ได้ดังนี้

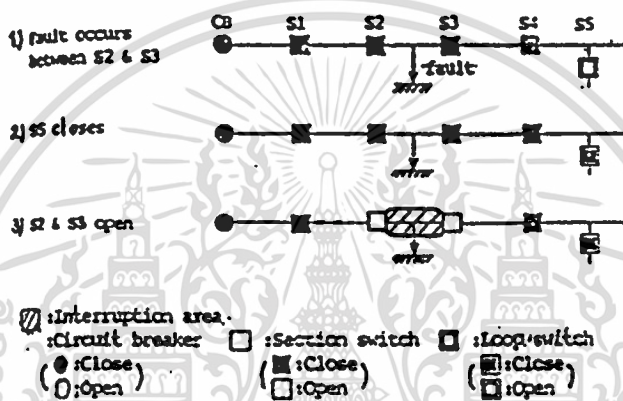
2.7.1 เกิด Ground Fault โดยทั่วไปแล้วกระแสฟอลต์ที่ลง Ground จะมีค่าระหว่าง 10 A. ถึง 1 kA. ในระบบไม่มี Ground และในระบบที่มีความต้านทานของสายดิน (Resistance Ground System) ส่วนที่เกิดฟอลต์จะถูกแยกออกโดยปราศจากการ Tripping ของ Circuit Breaker ในสถานีไฟฟ้าย่อย เพราะว่ากระแสฟอลต์ที่ลงดินสามารถทำให้เกิดปัญหาขึ้นในระบบสายดินได้ ด้วยเหตุนี้ในส่วนที่ไม่เกิดฟอลต์จึงไม่ถูกรบกวนโดยฟอลต์ ขบวนการแยกส่วนที่เกิดฟอลต์ออกสามารถแสดงได้ดังภาพที่ 28

เมื่อมีกระแสฟอลต์ลงดินเกิดขึ้นคอมพิวเตอร์ที่ศูนย์ควบคุมก็จะรวบรวมสัญญาณฟอลต์ที่เกิดขึ้นใน Line Senser โดยอาศัยระบบสื่อสาร และทำการตรวจจับฟอลต์ว่าเกิดขึ้นที่ใด จากนั้นคอมพิวเตอร์ที่ศูนย์ควบคุมก็จะส่งสัญญาณควบคุมไปยัง Remote Terminal Unit เพื่อทำการสั่ง Automatic Reclosing Sectionlizer ให้เปิดวงจรเป็นเวลา 140 ms. ดังภาพที่ 29 หลังจากนั้นคอมพิวเตอร์ที่ศูนย์ควบคุมก็จะส่งสัญญาณควบคุมไปเปิดวงจรด้านแหล่งจ่ายและด้านโหลดจากจุดที่เกิดฟอลต์ Automatic Reclosing Sectionlizer จะถูกเปิดออกเป็นเวลา 50 ms. เพื่อเป็นการป้องกันการเกิด Over Load ระหว่าง 2 Feeder ซึ่งการทำงานเหล่านี้จะต้องเสร็จสิ้นก่อนที่ Circuit Breaker จะทำการ Lock-Out ซึ่งระยะเวลาทั้งหมดจะต้องแปรผันตามจำนวนของตัว Remote Terminal Unit เช่น

ถ้าจำนวน Remote Terminal Unit เท่ากับ 100 ตัวจะต้องใช้เวลาในการแยกส่วนที่เกิดฟอลต์ออกจากระบบเป็นเวลา 700 ms. กระบวนการในการแยกฟอลต์ออกจากระบบนี้ต้องอาศัยสวิทช์ความเร็วสูง และระบบสื่อสารที่ดี

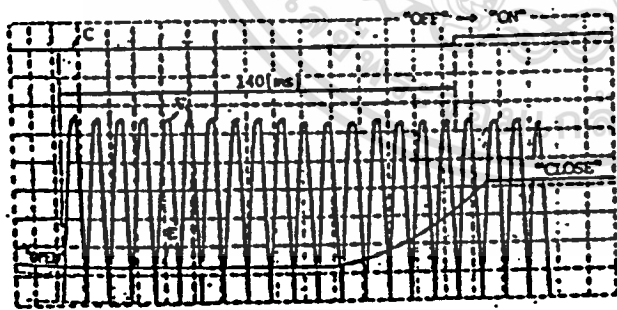
ภาพที่ 28

แสดง Isolation procedure on a ground fault

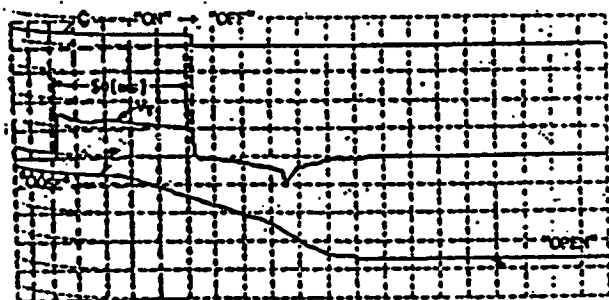


ภาพที่ 29

แสดงการ Closing และ Opening



(a) Closing



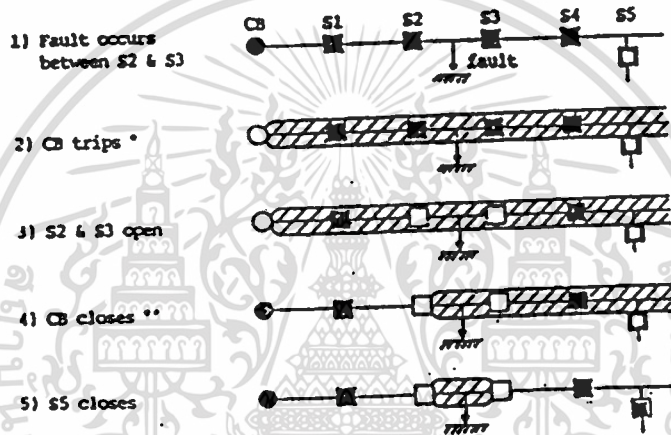
(b) Opening

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.7.2 เกิด Short Circuit Fault ฟอลท์ที่เกิดจากการลัดวงจรนี้ Automatic reclosing Sectionlizer ไม่สามารถที่จะตัดกระแสฟอลท์ชนิดนี้ได้ซึ่งส่วนใหญ่จะมีค่ามากกว่า 10 kA. ดังนั้น Automatic reclosing Sectionlizer ของส่วนที่เกิดฟอลท์จะถูกเปิดออกหลังจาก Circuit Breaker ทำการ Lock-Out ขบวนการในการแยกออกจากระบบนี้สามารถแสดงได้ดังภาพที่ 30

ภาพที่ 30

แสดง Isolation Procedure on a Short Circuit fault



เมื่อกระแสฟอลท์ที่เกิดจากการ Short Circuit ขึ้นในระบบจำหน่ายคอมพิวเตอร์ที่ศูนย์ควบคุมก็จะทำการตรวจจับและแสดงบริเวณที่เกิดฟอลท์ หลังจากที Circuit Breaker ทำการ Lock-Out Automatic Reclosing Sectionlizer ของทั้งสองด้านจะเปิดออกและ Circuit Breaker ก็จะปิดลงอีกครั้ง ดังนั้นจึงมีฟอลท์ที่เกิดขึ้นเพียงส่วนเดียว และการแยกฟอลท์ออกจากระบบก็ใช้เวลาเพียง 1 วินาที โดยยังไม่รวมระยะเวลาที่ Circuit Breaker ปิดลงอีกครั้ง

บทที่ 3

ไมโครคอมพิวเตอร์กับการอินเทอร์เน็ตเฟส

3.1 หลักการเบื้องต้นของ 8255

ไอซี 8255 เป็นไอซี 40 ขา ที่ได้รับการออกแบบมาเพื่อให้มีสัญญาณเชื่อมโยงกับ 8088 และ Z-80 แต่สัญญาณนี้เหมาะกับการใช้ร่วมกันกับ IBM PC ไอซี 8255 นี้เป็นไอซีที่ต่อเป็นพอร์ตให้กับ ไมโคร โปรเซสเซอร์ได้ 3 พอร์ต โดยมีโครงสร้างพื้นฐานดังภาพที่ 31



การเรียกพอร์ตของ 8255 จะเรียกพอร์ตว่า พอร์ต A,B และ C โดยพอร์ต C แยกเป็น 2 ส่วน คือ $PC_0 - PC_3$ เรียกว่า C ต่ำ จำนวน 4 บิตและพอร์ต C บน $PC_4 - PC_7$ ที่พิเศษคือ พอร์ตทักพอร์ตทุกพอร์ทเป็นได้ทั้งพอร์ตอินพุทและพอร์ตเอาต์พุท

3.1.1 ขาต่าง ๆ ของ 8255 เพื่อให้เข้าใจวิธีการต่อใช้งานต่าง ๆ ระหว่าง IBM PC/AT กับ 8255 จึงจำเป็นต้องเข้าใจความหมายและตำแหน่งของขาต่าง ๆ ทั้ง 40 ขาของไอซีซึ่งประกอบด้วย

$D_0 - D_7$, เป็นขาข้อมูลของอินพุท-เอาต์พุท ที่จะต้องผ่านเข้าออกจากส่วนนี้ $D_0 - D_7$ จึงต้องต่อเข้ากับระบบบัสของไมโคร โปรเซสเซอร์ เพื่อให้ไมโคร โปรเซสเซอร์สามารถอ่านหรือเขียนข้อมูลออกทางพอร์ตผ่านทางบัสนี้

CS ขานี้เป็นขาอินพุทที่จะรับสัญญาณจากภายนอกเพื่อเลือกชิป 8255 นี้โดยเมื่อขานี้เป็นลอจิก “0” จะทำให้ตัว 8255 ต่อเข้ากับระบบบัสของไมโครโปรเซสเซอร์ เพื่อให้ไมโครโปรเซสเซอร์เขียนหรืออ่านข้อมูลออกจากพอร์ตได้

RD ขาสัญญาณการอ่านเป็นสัญญาณอินพุทที่จะส่งออกมาจาก CPU เมื่อสัญญาณที่ขานี้เป็น “0” และ CS เป็น “0” ตัว 8255 จะทำให้ตัวชิพที่อ่านข้อมูลจากบัส ในขณะที่เป็นอินพุท

WR ขาสัญญาณการเขียน จะแอกทีฟเมื่อสัญญาณ WR เป็น “0” และ CS เป็น “0” สัญญาณนี้จะมาจาก CPU เมื่อต้องการเขียนข้อมูลลงบนพอร์ตที่กำหนด

$A_0 - A_1$ ขาแอดเดรสลอจิกของขาทั้งสองข้าง จะถอดรหัสเป็น 4 เพื่อกำหนดครีจิสเตอร์ภายในที่เชื่อมต่อกับพอร์ตอินพุท-เอาต์พุทของ 8255

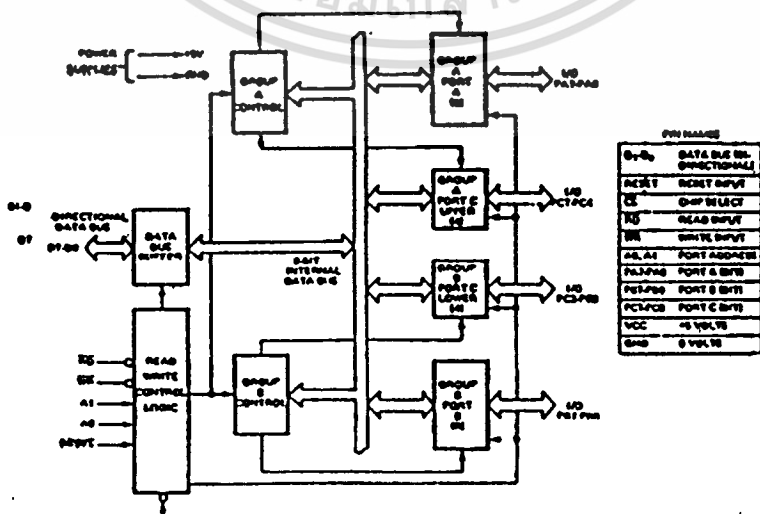
RESET ขานี้เป็นขาสัญญาณที่ส่งมาจากภายนอกเข้ามาทำการรีเซ็ต 8255 เพื่อเคลียร์สถานะต่าง ๆ ของ 8255 เมื่อ 8255 ได้รับการรีเซ็ต มันจะกลับเข้าสู่โหมดอินพุทหรือทุกพอร์ตเป็นพอร์ตอินพุท

$PA_0 - PA_7$ เป็นสายสัญญาณที่พอร์ตของ 8255 ที่ชื่อพอร์ต A การเลือกพอร์ตโดยขาแอดเดรส $A_0 - A_1$

$PC_0 - PC_1$ เป็นสายสัญญาณที่พอร์ต C ของ 8255 การกำหนดพอร์ตนี้จะได้รับการกำหนดโดยขาแอดเดรส $A_0 - A_1$ พอร์ต C นี้แบ่งเป็น 2 กลุ่ม คือ กลุ่ม $PC_0 - PC_7$ ดังภาพที่ 32

ภาพที่ 32

แสดงตำแหน่งการวางขาของ 8255



3.1.2 โหมดการทำงานของ 8255 การเลือกการทำงานของ 8255 สามารถให้ทำงานได้หลายรูปแบบ โดยเรียกแต่ละแบบว่าโหมด(MODE) ต่าง ๆ มีอยู่ 3 โหมด คือ โหมด 0 โหมด 1 และโหมด 2 ดังรายละเอียดที่จะอธิบายต่อไปนี้

3.1.2.1 โหมด 0 โหมดนี้เป็นโหมดที่กำหนดให้พอร์ตทุกพอร์ตบนตัว 8255 เป็นพอร์ตอินพุท-เอาต์พุท แบบพื้นฐาน รูปแบบความเป็นไปได้จึงมีทั้งสิ้น 16 รูปแบบ ตามลักษณะของพอร์ต A,B,C บน และ C ล่างลักษณะควบคุมแต่ละรหัสควบคุมแต่ละแบบจะเป็นดังภาพที่ 33 และภาพที่ 49 และสามารถแสดงสัญญาณของแต่ละพอร์ตได้ดังตารางที่ 3

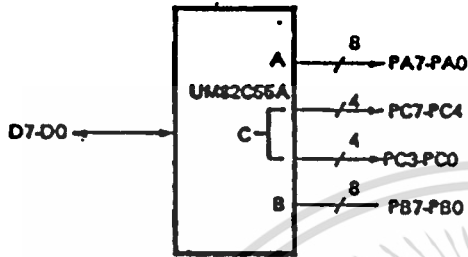
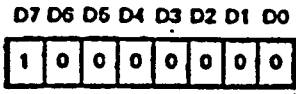
ตารางที่ 3
แสดงสัญญาณแต่ละพอร์ตของไอซี 8255

A		B		Group A		n	Group B	
D ₄	D ₃	D ₁	D ₀	Port A	Port C (Upper)		Port B	Port C (Lower)
0	0	0	0	OUTPUT	OUTPUT	0	OUTPUT	OUTPUT
0	0	0	1	OUTPUT	OUTPUT	1	OUTPUT	INPUT
0	0	1	0	OUTPUT	OUTPUT	2	INPUT	OUTPUT
0	0	1	1	OUTPUT	OUTPUT	3	INPUT	INPUT
0	1	0	0	OUTPUT	INPUT	4	OUTPUT	OUTPUT
0	1	0	1	OUTPUT	INPUT	5	OUTPUT	INPUT
0	1	1	0	OUTPUT	INPUT	6	INPUT	OUTPUT
0	1	1	1	OUTPUT	INPUT	7	INPUT	INPUT
1	0	0	0	INPUT	OUTPUT	8	OUTPUT	OUTPUT
1	0	0	1	INPUT	OUTPUT	9	OUTPUT	INPUT
1	0	1	0	INPUT	OUTPUT	10	INPUT	OUTPUT
1	0	1	1	INPUT	OUTPUT	11	INPUT	INPUT
1	1	0	0	INPUT	INPUT	12	OUTPUT	OUTPUT
1	1	0	1	INPUT	INPUT	13	OUTPUT	INPUT
1	1	1	0	INPUT	INPUT	14	INPUT	OUTPUT
1	1	1	1	INPUT	INPUT	15	INPUT	INPUT

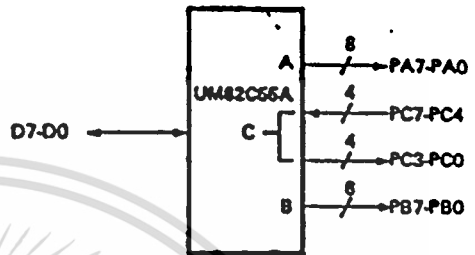
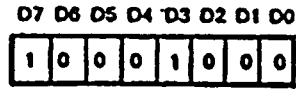
ภาพที่ 33

แสดงลักษณะการใช้งานของโหมด 0

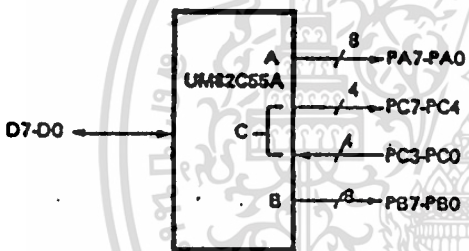
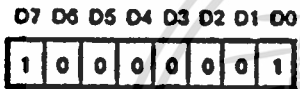
CONTROL WORD #0



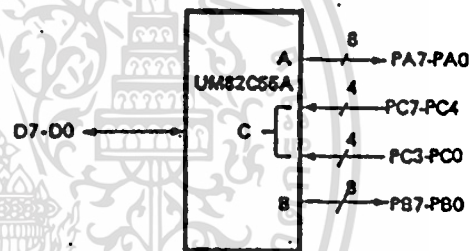
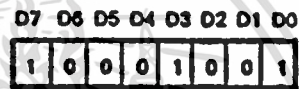
CONTROL WORD #4



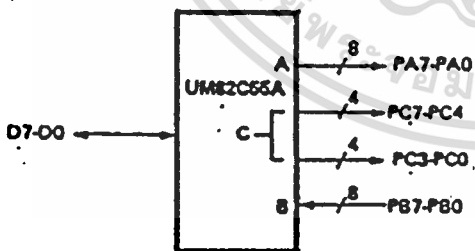
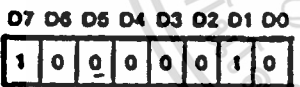
CONTROL WORD #1



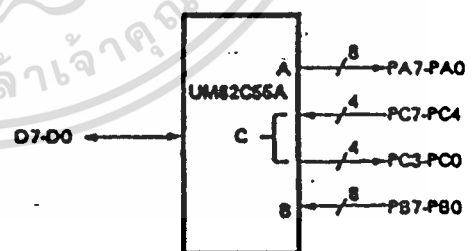
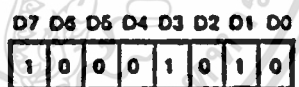
CONTROL WORD #5



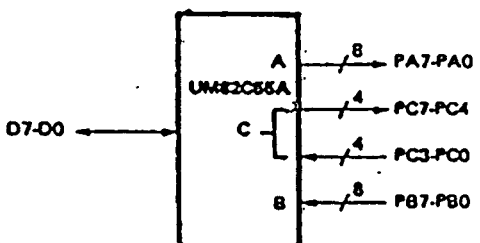
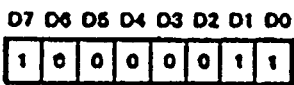
CONTROL WORD #2



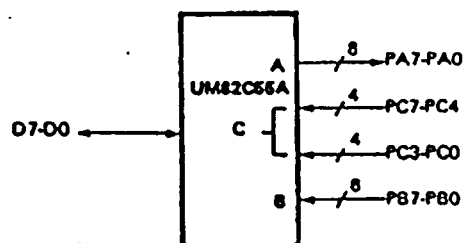
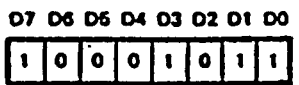
CONTROL WORD #6



CONTROL WORD #3



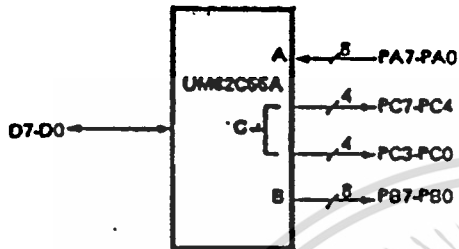
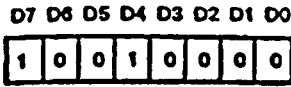
CONTROL WORD #7



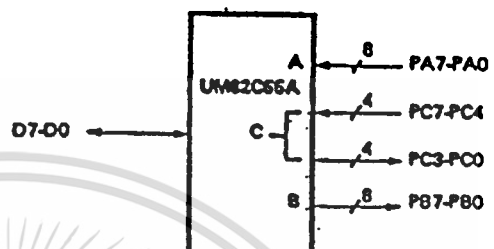
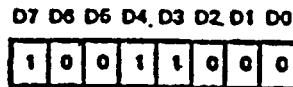
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แสดงลักษณะการใช้งานของโหมด 0 (ต่อ)

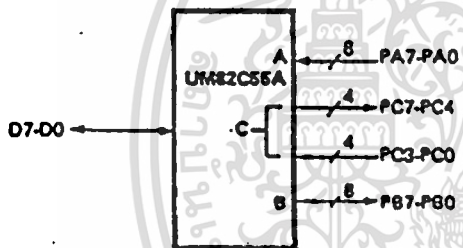
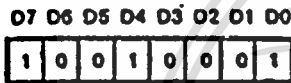
CONTROL WORD #8



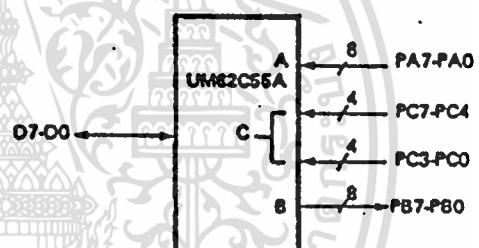
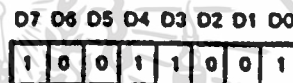
CONTROL WORD #12



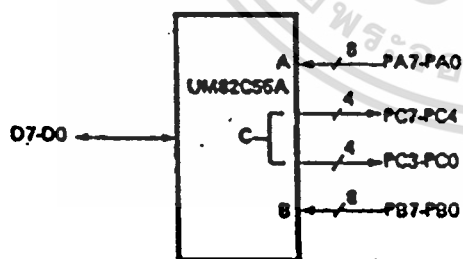
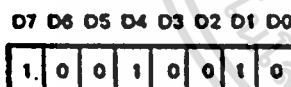
CONTROL WORD #9



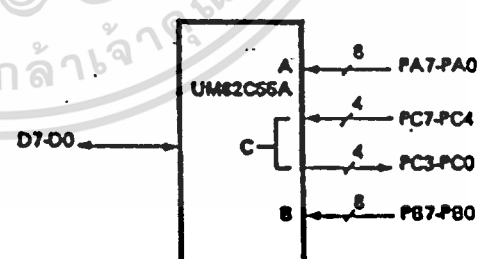
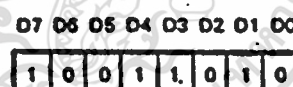
CONTROL WORD #13



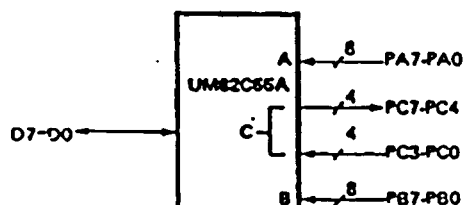
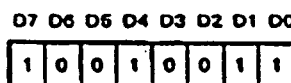
CONTROL WORD #10



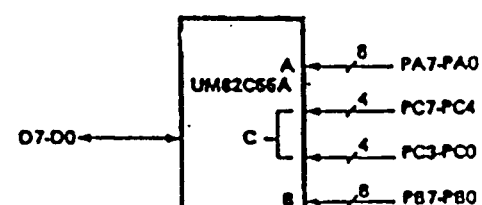
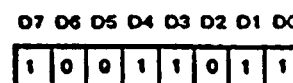
CONTROL WORD #14



CONTROL WORD #11



CONTROL WORD #15



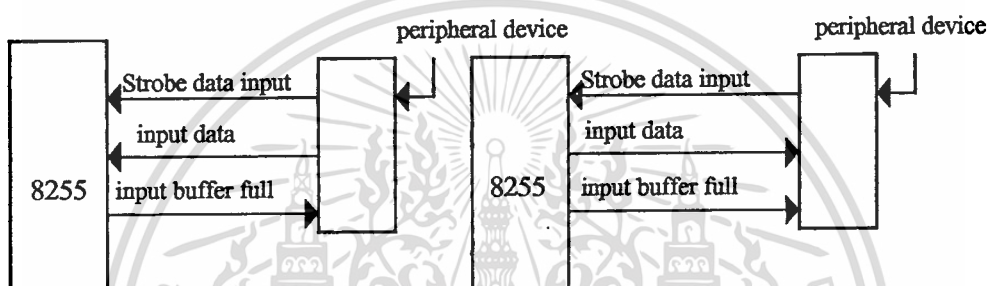
เอกสารนี้เป็นเอกสารที่ส่งมอบไปสำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้ทำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.1.2.2 โหมด 1 การทำงานของ 8255 ในโหมดที่ทำให้อินพุต-เอาต์พุต มีการตรวจสอบสัญญาณ (Handshaking) โดยใช้อินพุต-เอาต์พุตของพอร์ต A และ B เป็นหลักและใช้พอร์ต C บนเป็นสัญญาณ Handshake ของพอร์ต A ส่วนพอร์ต C ล่างเป็นสัญญาณ handshake ของพอร์ต B ดังภาพที่ 35 และตารางที่ 4

ภาพที่ 34

แสดงโครงสร้างของการจัดสัญญาณ Handshake



ตารางที่ 4

แสดงหน้าที่ต่าง ๆ ของพอร์ตของ C เป็นสัญญาณ Handshake เมื่อ 8255 ทำงานที่โหมด 1

ขา	กรณีอินพุต	กรณีเอาต์พุต
PC ₀	INTR _B	INTR _B
PC ₁	IBF _B	OBF _B
PC ₂	STB _B	ACK _B
PC ₃	INTR _A	INTR _A
PC ₄	STB _A	I/O
PC ₅	IBF _A	I/O
PC ₆	I/O	ACK _A
PC ₇	I/O	OBF _A

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แนวความคิดของการใช้ อินพุท-เอาต์พุท แบบ handshake ก็เพื่อให้มีการชิง โครนัสระหว่าง อุปกรณ์ภายนอกที่ทำงาน ได้ช้ากับการทำงานของคอมพิวเตอร์ที่ทำงาน ได้เร็ว เช่นเครื่องพิมพ์ทำงาน ได้ช้าเมื่อคอมพิวเตอร์ส่งตัวอักษรตัวแรกมาพิมพ์ เครื่องพิมพ์รับตัวอักษรและกำลังจะพิมพ์ คอมพิวเตอร์ก็ส่งตัวที่ 2 และตัวที่ 3 ตามมา ทำให้การประมวลผลของอุปกรณ์เครื่องพิมพ์ทำงานไม่ทันอาจทำให้ข้อมูลเสียหาย ดังนั้น เครื่องพิมพ์จึงส่งสัญญาณบอกคอมพิวเตอร์ว่า อย่างเพิ่งส่งมา เพราะยังไม่พร้อมรับ

ลักษณะของการรับส่งข้อมูล อินพุท-เอาต์พุท แบบมี Handshake แสดงได้ดังภาพที่ 34 และ PA₀ - PA₇ เป็นอินพุทโดยมีพอร์ต C เป็นสัญญาณ Handshake ดังภาพที่ 35 เมื่อโปรแกรม 8255 เป็น โหมด 1 แล้วตัว 8255 จะให้พอร์ต C เป็นสัญญาณควบคุมโดยแต่ละบิตของพอร์ต เป็นไปตามที่กำหนดไว้แล้วดังตารางที่ 3 โดยปกติ 8255 ยังให้สัญญาณอินเทอร์รัพต์ของ 8255 จะเกิดขึ้นที่ขา PC₀ และ PC₃ โดยที่บัฟเฟอร์พร้อมแล้ว และต้องการให้ซีพียูส่งอินพุท หรือเอาต์พุทมาที่บัฟเฟอร์ สัญญาณอินเทอร์รัพต์ที่เกิดขึ้น สังเกตว่าสัญญาณอินเทอร์รัพต์เป็นสัญญาณแอกทีฟลोजิก “1” ซึ่ง ตรงกับของ 8080 เมื่อใช้กับ IBM สัญญาณ INT ของ IBM จะรับด้วยลोजิก “0” โครงสร้างการ Handshake ของ 8255 แสดงด้วยสัญญาณทางไฟฟ้า ได้ดังภาพที่ 36 สังเกตว่าการทำงานของ 8255 เกี่ยวกับสัญญาณ RD และ WR ซึ่งจะทำให้สัญญาณควบคุมเปลี่ยนแปลงไป

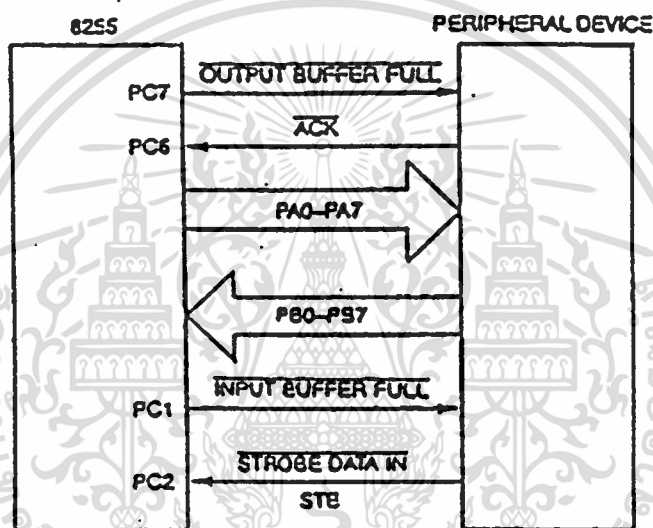
การตรวจสอบสัญญาณซึ่งกันและกันนี้ เป็นวิธีการรับส่งที่มีประสิทธิภาพ เช่น ในกรณีอินพุทเมื่ออุปกรณ์ภายนอกต้องการส่งข้อมูลให้ซีพียูก็จะส่งข้อมูลแบบขนานเข้ามา พร้อมทั้ง STD บอกร 8255 ตัว 8255 จะนำข้อมูลนั้นไปเก็บไว้ในรีจิสเตอร์ภายในก่อน แล้วส่งสัญญาณตอบบอกว่บัฟเฟอร์เต็มอยู่ณะ (IBF) อย่างเพิ่งส่งมาอีก ครั้นเมื่อซีพียูอ่านข้อความจากรีจิสเตอร์ไปแล้วส่วนของบัฟเฟอร์อินพุท (IBF) ก็จะบอกว่างแล้วส่งมาได้ อุปกรณ์ภายนอกก็จะส่งมาให้อีก

ทำนองเดียวกัน สำหรับพอร์ตเอาต์พุท เมื่อซีพียูส่งข้อมูลออกมาทางพอร์ตเอาต์พุทให้กับ 8255 ตัว 8255 ก็จะรับไว้ในรีจิสเตอร์ภายใน พร้อมสัญญาณออกไปบอกอุปกรณ์ภายนอกว่าเอาต์พุทบัฟเฟอร์มีสัญญาณอยู่ (OBF) อุปกรณ์ภายนอกเมื่อทราบและพร้อมจะอ่านก็จะส่งสัญญาณตอบรับ (ACK) พร้อมกับอ่านข้อมูลไป โดยสัญญาณ ACK จะมีความหมายว่าข้อมูลได้อ่านไปแล้ว ตัว 8255 ก็จะตอบกลับว่า บัฟเฟอร์ยังว่างอยู่และจะมีข้อมูลใหม่ส่งเข้ามาอีก

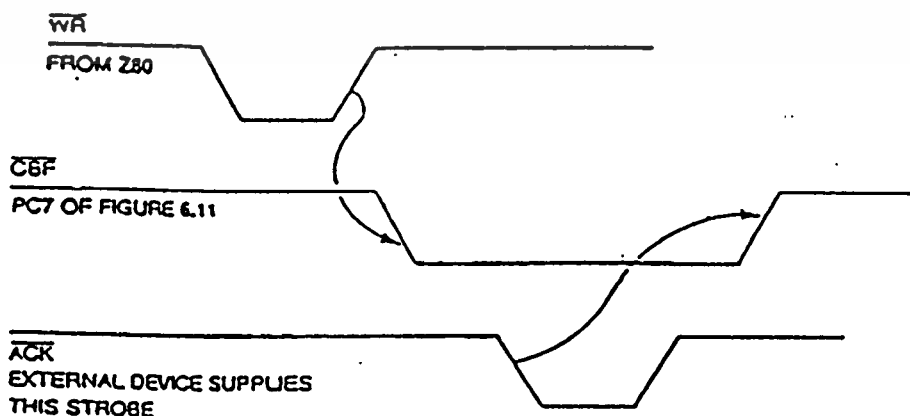
ในกรณีที่จะโปรแกรมโหมด 1 นั้น เราจะให้รหัสควบคุมเป็น 101 (I/O) 01 (I/O) 0 ในส่วน (I/O) หมายถึงถ้าเป็นอินพุทก็เป็น “1” เอาต์พุทก็เป็น “0” โดย I/O ตัวแรกเป็นของพอร์ต A ตัวที่ 2 พอร์ต B เช่นถ้าต้องการให้พอร์ต A เป็นเอาต์พุท และพอร์ต B เป็นอินพุท เราจะใช้รหัสควบคุมเป็น 10100110 หรือ A6H พิจารณาการทำงานของซีพียูจะเห็นว่า ทำอย่างไรจึงเขียนหรืออ่าน พอร์ต ได้ถูกต้องวิธีง่ายวิธีหนึ่งคือซีพียูจะคอยตรวจสอบสัญญาณของ 8255 เช่น กรณีเอาต์พุทซีพียู

จะคอยอ่านพอร์ต C แล้วตรวจสอบบิต 7 (OBF) หลังจากที่ส่งข้อมูลไปแล้ว ถ้าบิต 7 ยังเป็น “0” แสดงว่ายังไม่ได้รับการ STB แต่ถ้าเป็น “1” แล้วแสดงว่าอุปกรณ์ภายนอกได้รับข้อมูลไปแล้ว สำหรับกรณีอื่นทุกคอยตรวจสอบจากสัญญาณ IBF ได้เช่นกันว่า มีข้อมูลใหม่เข้ามาหรือยังคือตรวจสอบบิต PC₁ ของพอร์ต C

รูปที่ 35
แสดงการต่อ 8255 ในโหมด



ภาพที่ 36
แสดงช่วงเวลาการรับ / ส่งข้อมูลแบบ Handshake



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.1.2.3 โหมด 2 8255 ยังมีโหมดทำงานอีกโหมดคือ โหมด 2 ซึ่งทำได้เฉพาะพอร์ต A ในโหมดนี้ 8255 จะใช้พอร์ต A ทำหน้าที่เป็นพอร์ตแบบ 2 ทิศทางคือสามารถเป็นได้ทั้งอินพุตและเอาต์พุต โดยโครงสร้างของพอร์ต A ทั้งอินพุตมี Handshake ทั้งคู่ส่วนพอร์ต C ที่ใช้จะทำหน้าที่เป็นสัญญาณตรวจสอบแต่ละขาดังตารางที่ 5

ตารางที่ 5
แสดงหน้าที่ของพอร์ต C ในโหมด 2

PORT C LINE	DEFINITION
PC0	I/O
PC1	I/O
PC2	I/O
PC3	INTRA
PC4	STBA
PC5	IBF _a
PC6	ACCA
PC7	CBF _a

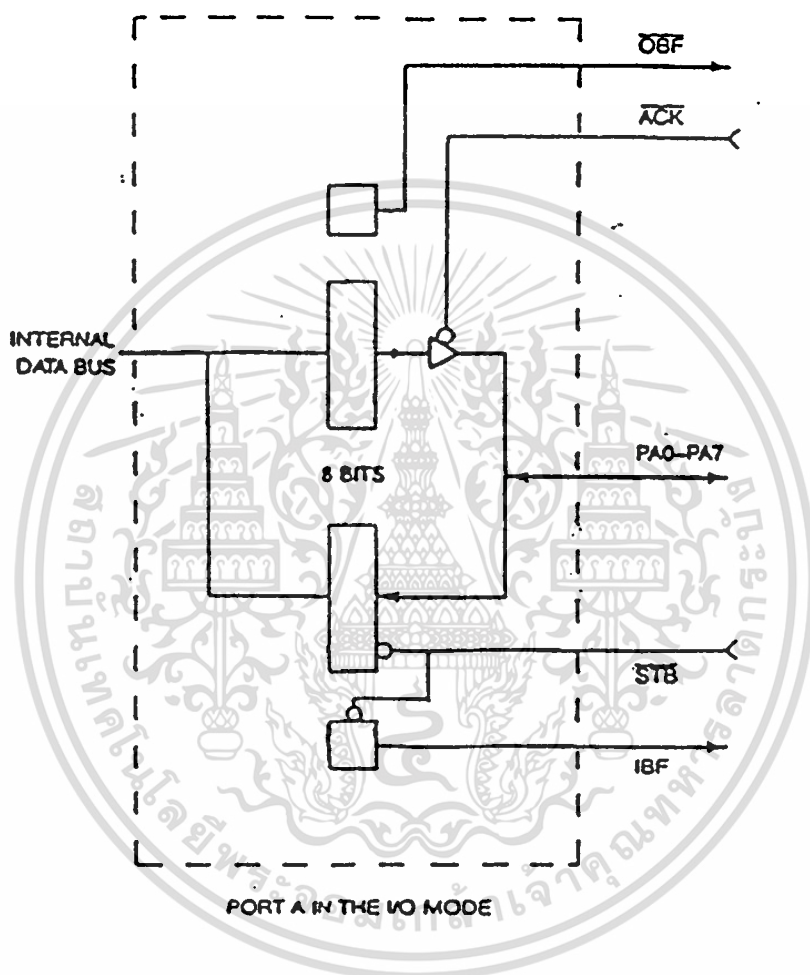
และจากภาพที่ 37 เป็นโครงสร้างของพอร์ต A ที่ทำงานแบบสองทิศทาง เมื่อโปรแกรมพอร์ต A เป็นโหมด 2 แล้ว พอร์ต B จะต้องโปรแกรมเป็นโหมด 0 หรือโหมด 1 ก็ได้ ซึ่งการทำงานได้แยกอิสระอีกในการใช้งานพอร์ตแบบ 2 ทิศทางนี้มีใช้ได้กับงานบางประเภท เช่น ใช้ในการรับส่งข้อมูลของพอร์ตมาตรฐานบางประเภท เช่น IEEE 488 หรือ ใช้เชื่อมโยงระหว่างคอมพิวเตอร์ในการรับส่งข้อมูลสลับกันไปกลับ

ในการทำงานของโหมด 1 หรือโหมด 2 สัญญาณควบคุมสามารถส่งผ่าน Interruptrequest เข้าไปยัง CPU โดยสัญญาณนี้สร้างขึ้นโดยพอร์ต C เพื่อจะไปทำการ Inhibited หรือ Enabled โดยการเซตหรือการรีเซตขา Inte F/F ของ CPU โดยสัญญาณ BIT SET / RESET ของพอร์ต C

BIT SET จะตอบสนองการอินเตอร์รัพต์ (INTERRUPT ENABLE)

BIT RESET จะไม่ตอบสนองการอินเตอร์รัพต์ (INTERRUPT DISABLE)

ภาพที่ 37
แสดงโครงสร้างของพอร์ต A ที่ทำงานแบบพอร์ต 2 ทิศทาง



3.2 อุปกรณ์ต่าง ๆ ที่ช่วยในการดีโค๊ดสัญญาณบน IBM

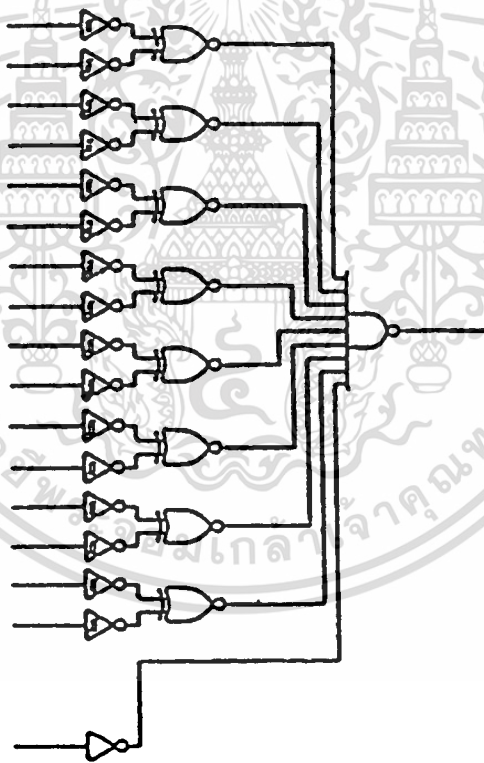
เมื่อทราบถึงสัญญาณต่าง ๆ ที่ 8255 ต้องการและเทคนิคการดีโค๊ดแบบต่าง ๆ ที่สามารถจะนำไปใช้ได้แล้วการที่จะต่อ 8255 เข้ากับเครื่องคอมพิวเตอร์ IBM PC/AT นั้นก็จะสะดวกขึ้นเพียงแต่ชุดทดลองเลือกแอดเดรสให้แก่ 8255 เท่านั้น และเพื่อให้การใช้งานได้สะดวกจึงใช้การดีโค๊ดแบบสะดวกเปลี่ยนแอดเดรสได้ด้วย ก่อนอื่นจะต้องทราบอุปกรณ์ต่าง ๆ ที่จะช่วยในการดีโค๊ด

เอกสารนี้เป็นเอกสารทสงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น เมื่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2.1 IC 74LS688 ICตัวนี้ทำหน้าที่ในการเปรียบเทียบสัญญาณ 2 สัญญาณว่าเหมือนกันหรือไม่ (“0” หรือ “1”) โดยสามารถเปรียบเทียบได้พร้อมกัน 8 คู่ โดยอินพุตแต่ละคู่จะอยู่ที่ขา P และ G และมีสัญญาณที่ขาอินพุตแต่ละคู่ตรงกัน จะทำให้มีสัญญาณออกมาที่อินพุตเป็นลอจิก “0” เมื่อมีอินพุตคู่ใดคู่หนึ่งมีสัญญาณแตกต่างกันเอาพุตจะออกเป็น “1” ทั้งนี้และ IC 74LS688 นั้นยังมีสัญญาณ ENABLE อีก 1 ขาคือ G ซึ่งจะควบคุมการทำงานของ IC74LS688 เมื่อขา G นั้นลอจิกเป็น “0” จึงจะมีการเปรียบเทียบสัญญาณต่าง ๆ ได้ ดังภาพที่ 38

ภาพที่ 38

แสดงการจัดขาและตารางเอาต์พุตของ IC 74LS688



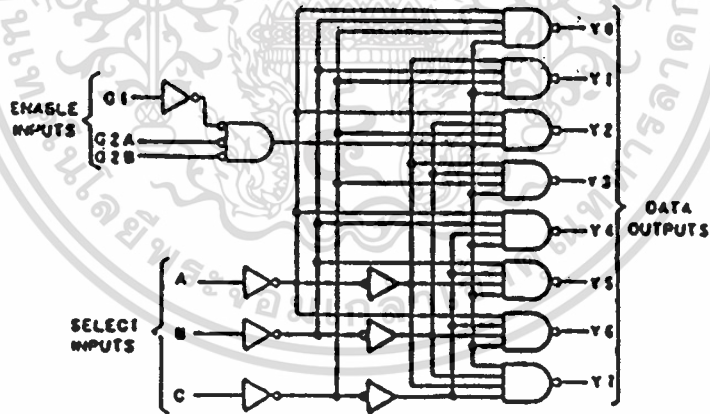
3.2.2 IC 74LS138 เป็น IC 3-LINE to 8-LINE DECEDER คือมีอินพุต 3 ตัวและมีเอาต์พุต 8 ตัว การให้เอาต์พุตจะให้ป็นสัญญาณ “0” ออกมาตามตารางที่ 7 และการจัดขาของ IC 74LS138 ดังภาพที่ 39

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 6
แสดงเอาต์พุตของ IC 74LS138

INPUTS					OUTPUTS							
ENABLE		SELECT			Y0	Y1	Y2	Y3	Y4	Y5	Y6	Y7
G1	G2 ^a	C	B	A								
X	H	X	X	X	H	H	H	H	H	H	H	H
L	X	X	X	X	H	H	H	H	H	H	H	H
H	L	L	L	L	L	H	H	H	H	H	H	H
H	L	L	L	H	H	L	H	H	H	H	H	H
H	L	L	H	L	H	H	L	H	H	H	H	H
H	L	L	H	H	H	H	H	L	H	H	H	H
H	L	H	L	L	H	H	H	H	L	H	H	H
H	L	H	L	H	H	H	H	H	L	H	H	H
H	L	H	H	L	H	H	H	H	H	L	H	H
H	L	H	H	H	H	H	H	H	H	H	L	H

ภาพที่ 39
แสดงการจัดขาของ IC 74LS138



3.3 การต่อเชื่อม 8255 กับ IBM PC/AT

พิจารณาขาต่าง ๆ ของ 8255 จะเห็นว่าส่วนขาควบคุมที่จะเชื่อมโยงเข้ากับบัสของ IBM นั้น สามารถเชื่อมโยงเข้ากับบัสโดยตรง การที่จะต่อ 8255 เข้ากับเครื่อง IBM PC/AT นี้จะต้องนำเอา สัญญาณต่าง ๆ ที่จะมาควบคุม 8255 ในการทำงานตามต้องการ สัญญาณที่ต้องใช้คือ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- A_0-A_1 เป็นสายแอดเดรสที่จะนำมาจากสล็อตของ IBM ไม่ต้องผ่านรีจิสเตอร์ดีโค้ดเพราะจะนำมาดีโค้ดบน 8255 ซึ่งการเปลี่ยนแปลงของ A_0 และ A_1 จะไปควบคุมรีจิสเตอร์ภายใน 8255 ซึ่งใช้ในการเลือกโหมดการทำงานต่าง ๆ ของ 8255 ซึ่งในการจะควบคุมให้ 8255 ทำงานในโหมดไหนนั้นต้องส่ง Control Word ไปที่คอนโทรลรีจิสเตอร์ของ 8255 ก่อน

- RESET เป็นสายสัญญาณที่จะนำมาจากสล็อตของ IBM ไม่ต้องผ่านวงจรใด ๆ เพราะ 8255 ต้องการสัญญาณ Reset ที่มีลอจิกเป็น “1” ซึ่งสัญญาณ Reset ของ IBM ก็มีลอจิก “1” ตรงกับ Reset ของ 8255 จึงสามารถต่อ Reset ของ IBM เข้ากับขา Reset ของ 8255 ได้โดยตรง

- RD สัญญาณ RD ของ 8255 ต่อเข้าโดยตรงกับสัญญาณ RD ของ IBM โดยสัญญาณนี้จะเป็น Active เมื่อเครื่องคอมพิวเตอร์ IBM ต้องการอ่านข้อมูลจากอุปกรณ์ภายนอกผ่านทาง 8255

- WR ซึ่งต่อเข้ากับขา WR ของ 8255 โดยตรงจะเป็น Active เมื่อขา WR มีลอจิก “0” คือต้องการเขียนข้อมูลผ่านบน 8255

- D_0-D_7 เป็น Data Bus ของ IBM ซึ่งต่อเข้ากับ D_0-D_7 ของ 8255 ได้โดยตรงซึ่งเป็นเส้น Data ที่ใช้ในการรับส่งข้อมูลต่าง ๆ ระหว่างเครื่อง IBM กับ 8255 ซึ่งที่ขา Data ทุกเส้นจะต้องต่อทางด้าน Pull Up เพื่อให้ข้อมูลที่ส่งมาไม่ให้เกิดความผิดพลาด

- AEN ต่อเข้ากับชุดดีโค้ด จากการใช้งานที่ต่อใช้งานกับ 8255 AEN จะต้องมีลอจิกเป็น “0”

- A_2-A_7 ต่อเข้ากับชุดดีโค้ดเพื่อให้สามารถเลือกแอดเดรสให้กับ 8255 ทำงานและสามารถเปลี่ยนแปลงแอดเดรสได้โดยเปลี่ยนค่าที่ Dip Switch

- CS ของ 8255 จะต่อเข้ากับชุดดีโค้ดเพื่อเลือกให้ 8255 ทำงานโดยถ้าขา CS มีลอจิก “0” Data Bus ของ 8255 จะต่อเข้ากับ Data Bus ของ IBM จึงสามารถอ่านและเขียนข้อมูลบน 8255 ได้ ซึ่งวงจรที่สมบูรณ์จะดีโค้ดให้ 8255 Active เมื่อ Y_0 ส่งสัญญาณลอจิก “0” มาที่ขา CS ของ 8255 ซึ่ง 8255 จะเริ่มที่แอดเดรส 300H

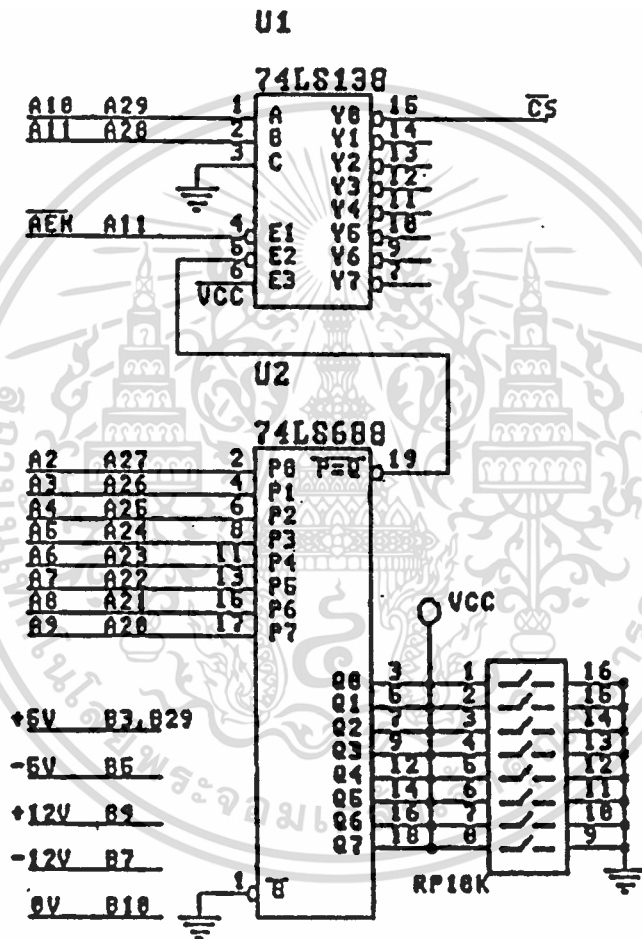
การเชื่อมต่อเครื่อง ไมโครคอมพิวเตอร์กับอุปกรณ์ภายนอก เป็นการทำงานขณะที่ติดต่อกับอุปกรณ์ภายนอกโดยผ่าน Parallel Port และสัญญาณต่าง ๆ บน Slot AT ซึ่งจะต้องออกแบบวงจรทางด้าน Hard Ware เพื่อนำมาใช้งานจริงโดยแยกออกเป็นส่วนตัวต่าง ๆ ดังนี้

3.3.1 วงจรดีโค้ด สัญญาณที่จะนำมาทำการดีโค้ดส่วนใหญ่ก็คือ สัญญาณจากแอดเดรสบัส ส่วน AEN ที่เลือกมาใช้เพราะเป็นสัญญาณของ IBM PC ที่บอกให้อุปกรณ์ภายนอกรู้ว่านี่เป็นแอดเดรสที่ติดต่อกับอุปกรณ์ I/O เมื่อ AEN เป็น “0” และเมื่อ AEN เป็น “1” เป็นขบวน DMA ของ IBM จากวงจรสัญญาณที่นำมาดีโค้ดได้แก่ A_2-A_{11} เพราะ A_0 และ A_1 นำไปดีโค้ดที่ 8255 แล้ว ส่วนบิตที่สามารถเซตได้จากวงจรได้แก่ $A_2 - A_7$ ส่วน A_8 และ A_9 นั้นต้องเป็น “1” อยู่เสมอ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

และในส่วนของ A_{10} - A_{11} เป็นตัวดีโค้ดให้ IC 74LS138 ส่งสัญญาณ CS ไปให้แก่ 8255 แสดงได้ดังภาพที่ 40

ภาพที่ 40
แสดงวงจรดีโค้ดเคอร์



การทำงานของวงจรดีโค้ด เริ่มแรกเมื่อสัญญาณจากแอดเดรสบิต A_2 - A_9 เข้ามาสู่ IC 74LS688 ซึ่งเป็นตัวเปรียบเทียบสวิตช์ที่ตั้งไว้ว่าตรงกันหรือไม่ เมื่อตรงกันทุกบิต IC 74LS688 ก็จะส่งพัลส์มาจากขา 19 ซึ่งจะมีลอจิกเป็น "0" ไปให้แก่ IC 74LS138 ที่ขาสัญญาณ G_2A เมื่อ G_2A เป็น "0" สัญญาณ G_2A (จาก IBM PC คือ EN) เป็น "0" และ G เป็น "1" A, B, C เป็น "0" หมค IC 74LS138 ก็จะส่งสัญญาณ CS ออกมาที่ขา 15 (Y_0) ถ้าที่ $A = "1"$ B และ C จะ = "0" IC 74LS138 จะส่งสัญญาณ CS ออกมาที่ขา 14 คือ Y_1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.3.2 วงจรอินเทอร์เฟซ 8255 กับ IBM PC/AT พิจารณาจากขาต่าง ๆ ของ 8255 จะเห็นว่าส่วนขาควบคุมที่จะเชื่อมโยงเข้ากับบัสของ IBM นั้นสามารถเชื่อมโยงเข้ากับบัสโดยตรง ซึ่งการนำสัญญาณต่าง ๆ ที่จะมาควบคุม 8255 ให้ทำงานตามต้องการนั้นสัญญาณที่ต้องใช้ประกอบด้วย

- A_0-A_1 ซึ่งไม่ต้องผ่านวงจรถักเพื่อเลือกพอร์ตต่าง ๆ ของ 8255
- Reset 8255 จะต่อเข้ากับขา Reset ของ IBM ของ 8255 ได้โดยตรง
- RD ของ 8255 ต่อเข้ากับสัญญาณ RD ของ IBM ได้โดยตรง
- WR ของ 8255 ต่อเข้ากับสัญญาณ WR ของ IBM ได้โดยตรง
- D_0-D_7 เป็น Data Bus ของ IBM ซึ่งต่อเข้ากับ D_0-D_7 ของ 8255 ได้โดยตรงซึ่งเป็นเส้น Data ที่ใช้ในการรับส่งข้อมูลต่าง ๆ ระหว่างเครื่อง IBM กับ 8255 ซึ่งที่ขา Data ทุกเส้นจะต้องต่อทางด้าน Pull Up เพื่อให้ข้อมูลที่ส่งมาไม่เกิดความผิดพลาด
- AEN ต่อเข้ากับชุดชิคโค้ด จากการใช้งาน AEN จะต้องมีลอจิกเป็น "0"
- A_2-A_9 ต่อเข้ากับชุดชิคโค้ดเพื่อให้สามารถเลือกแอดเดรสให้กับ 8255
- CS จะต่อเข้ากับชุดชิคโค้ดเพื่อเลือกให้ 8255 ทำงานที่แอดเดรส 300H

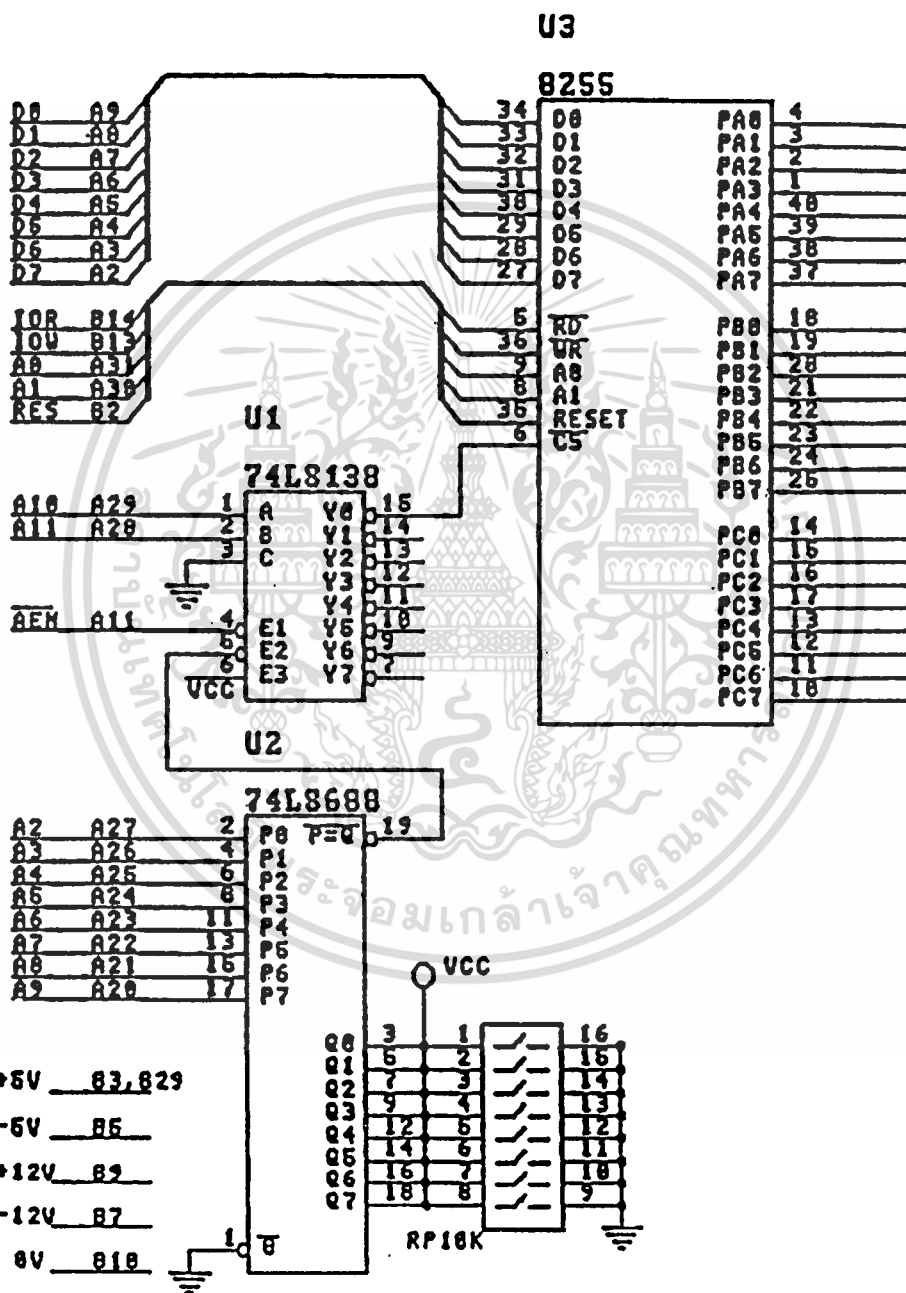
วงจรอินเทอร์เฟซอุปกรณ์ภายนอกกับเครื่อง ไมโครคอมพิวเตอร์ ได้แก่ IC 8255 IC 74LS688 และ IC 74LS138 โดยออกแบบให้สามารถทำงานได้เมื่อใช้ร่วมกับ Slot AT ดังภาพที่ 41

3.3.3 วงจรแปลงสัญญาณอนาล็อกเป็นดิจิตอล สำหรับกระบวนการ ADC บนการ์ด จะเป็นหน้าที่ของ IC ชิปปอร์ ADC 0804 ซึ่งจะเป็นชิปแปลงสัญญาณอนาล็อกเป็นดิจิตอลขนาด 8 บิต สัญญาณอินพุตให้ IC ADC 0804 คือสัญญาณ CS จะได้มาจากการ PC, ของ 8255 ทั้งนี้เพื่อหลีกเลี่ยงการทำงานซ้ำซ้อน โดยให้อยู่ในโหมด Free Run สัญญาณ RD, INTR, WR และ CS ต่อเพื่อควบคุมการทำงานของ IC ADC 0804 แรงดันอ้างอิงของ IC ADC 0804 ถูกกำหนดด้วยค่าแรงดันอ้างอิงส่วนสองจากภายนอกโดย R_2 และ R_3 ซึ่งได้ค่า 2.5 V. ส่วนแรงดัน $VI+$ ก็เช่นกันคือได้จากภายนอกและ $VI-$ ต่อกาวด์ เอ้าต์พุตของ IC ADC 0804 ออกมาที่ขา DB_0-DB_7 , ต่อเข้ากับบัสข้อมูล PA_0-PA_7 , ผ่าน 8255 ได้สัญญาณ D_0-D_7 , ป้อนเข้าบัสของ PC ต่อไป

ส่วนการเลือกอินพุตใช้ IC CD 4051 เป็นตัวเลือก ซึ่งสามารถควบคุมที่ขาสัญญาณ A, B และ C ซึ่งต่ออยู่กับ PC_0, PC_1 และ PC_2 ของ 8255 ตามลำดับ ซึ่งวงจรแปลงสัญญาณอนาล็อกเป็นดิจิตอล และรวมถึงการเป็นอินพุตทั้ง 3 แชนแนลซึ่งได้แก่ IC ADC 0804 และ IC CD 4051 ดังภาพที่ 42

ภาพที่ 41

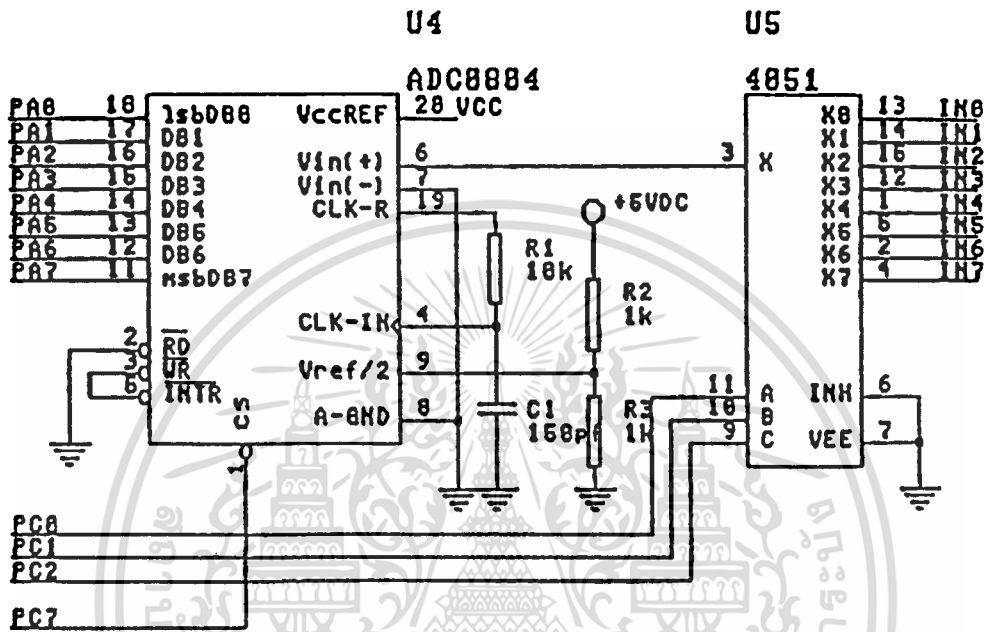
แสดงวงจรที่สมบูรณ์ของแผ่นวงจรอินเทอร์เฟซด้วย 8255 บนเครื่อง IBM PC



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดก็ตาม ห้ามนำไปใช้เพื่อการค้า และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาพที่ 42

แสดงวงจรแปลงสัญญาณอนาล็อกเป็นดิจิทัล



3.4 ความรู้ทั่วไปเกี่ยวกับภาษาซี

ภาษาซี เป็นภาษาระดับกลาง(Middle-Language) ที่นำมาเขียนโปรแกรมระบบปฏิบัติการ ยูนิกซ์ (มนตรี พจนารถาวณิช 2537:15-38) ภาษาซีมีความสามารถเหมือนกับภาษาแอสเซมบลี คือ มีคำสั่งที่สามารถเข้าถึงบิต ไบต์ และในตำแหน่งของหน่วยความจำของเครื่องได้ แต่เขียนง่ายเหมือนกับภาษาระดับสูงทั่วไป เช่น ภาษาบาสกาล หรือเบสิก เป็นต้น นอกจากนั้นโปรแกรมที่เขียนด้วยภาษาซียังมีข้อดีที่สำคัญอีกประการหนึ่งคือภาษาซีที่ใช้บนเครื่องหนึ่งยังสามารถใช้บนเครื่องอื่น ๆ ได้โดยไม่ต้องดัดแปลงและมีลักษณะเป็นภาษาโครงสร้างที่เหมาะสมจะใช้ในการพัฒนาโปรแกรม โดยมีความเร็วสูงในการทำงาน จึงทำให้ภาษาซีถูกนำมาใช้ในการเขียนโปรแกรมสำเร็จรูปต่าง ๆ เช่น DEBASEIV , Microsoft Excel , Sidekick , Oracal , SAS , CU Writer เป็นต้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาษาซีที่ใช้ในงานวิจัยของวิทยานิพนธ์นี้ จะอ้างถึงเทอร์โบซีเป็นหลักเพราะสามารถใช้งานเครื่องไมโครคอมพิวเตอร์ได้ ทั้งรุ่น XT , AT และ PS/2 ที่มีหน่วยขับแผ่นบันทึก (Floppy Disk Drive) ตั้งแต่ 1 หน่วยขับ (Drive) ขึ้นไปหรืองานบันทึกแบบแข็งจะสะดวกและรวดเร็วกว่าในการทำงานขนาดของหน่วยความจำไม่ควรต่ำกว่า 448 kB โดยอุปกรณ์ที่มีอยู่นี้ทั่วไปในไมโครคอมพิวเตอร์จะมีใช้กันอยู่ นอกจากนั้นเทอร์โบซียังหาได้ง่าย

3.5 การเขียนโปรแกรมด้วยกราฟฟิกเทอร์โบซี

ลักษณะการแสดงผลในโหมดกราฟฟิกนั้น จะมีลักษณะเป็นจุดมาประกอบกันเป็นรูปภาพจำนวนใน 1 จอภาพ อาจจะมีขนาดตั้งแต่ $320 * 200 * 640 * 768$ ก็ได้ ขึ้นอยู่กับคุณภาพของวงจรที่ใช้ในการแสดงผล ถ้าต้องการความละเอียดของภาพที่แสดงผลมาก ก็ต้องมีจำนวนจุดทางแนวและคอลัมน์สูงด้วย การอ้างอิงตำแหน่งบนจอภาพในขณะที่อยู่ในโหมดกราฟฟิกจะต้องอ้างอิงตำแหน่งเป็นแนวเป็นคอลัมน์เหมือนกับ การอ้างอิงในเท็กซ์โหมดแต่จุดเริ่มต้นของตำแหน่งจะอยู่แถวที่ 0 คอลัมน์ที่ 0 แทนที่จะเป็น แถวที่ 1 คอลัมน์ที่ 1 เหมือนเท็กซ์โหมด ดังนั้นผู้เขียนโปรแกรมในโหมดกราฟฟิก ควรจะต้องรู้ว่าวงจรที่ใช้ในการแสดงผลในเครื่องคอมพิวเตอร์เป็นของบริษัทใดและถ้าสามารถแสดงผลเป็นแบบสี หรือขาวดำ ด้วยเพราะในเทอร์โบซีจะมีไฟล์ไดรเวอร์อยู่หลายไฟล์เพื่อเตรียมไว้สำหรับวงจรแสดงผลแบบต่าง ๆ ดังตารางที่ 7

ตารางที่ 7

แสดงระบบกราฟฟิกที่ใช้กับเทอร์โบซี

ไดรเวอร์	สำหรับ
ATT.BGI	AT&T Graphics
CGA.BGI	CGAและMCGA Graphics
EGA VGA.BGI	EGAและVGA Graphics
HERC.BGI	Hercules Graphics
IBM 8514.BGI	IBM 8514 Graphics
PC 3270.BGI	IBM 3270 Graphics

จากตารางที่ 7 ในคอลัมน์ทางซ้ายมือจะเป็นไฟล์ประเภท BGI (Borland Graphics Interface) ซึ่งเป็นไฟล์เก็บคำสั่งสำหรับใช้กับระบบกราฟฟิก ที่มีชื่ออยู่ในคอลัมน์ทางขวามือ ไฟล์ชนิดนี้นิยมเรียกว่า ไคร์เวอร์ หรือตัวกระตุ้นให้มีการแสดงกราฟฟิกที่จอภาพ เนื่องจากระบบมีหลายชนิดแต่ละชนิดมีรายละเอียดแตกต่างกัน จึงเป็นเหตุให้ต้องมีไคร์เวอร์หลายไฟล์ คือ ไฟล์ CGA.BGI ใช้กับกราฟฟิกชนิด CGA ไฟล์ EGA VGA.BGI ใช้กับกราฟฟิกชนิด EGA และ VGA เป็นต้น

ฟังก์ชันที่ใช้ในโหมดกราฟฟิกจะเก็บอยู่ในไฟล์ชื่อ Graphics.h ดังนั้น ถ้าจะใช้ฟังก์ชันในโหมดเหล่านี้ จะต้อง # include ไฟล์ Graphics.h เข้ามาก่อนตอนต้นของโปรแกรม ฟังก์ชัน initgraph เป็นฟังก์ชันที่ใช้เพื่อต้องการเข้าสู่โหมดกราฟฟิกโดยฟังก์ชันนี้จะทำหน้าที่เป็นไคร์เวอร์ที่อยู่ใน disk มาเก็บไว้ในหน่วยความจำ ซึ่ง driver เป็นตำแหน่งของตัวแปรจำนวนเต็ม ที่ทำหน้าที่กำหนดชนิดของไฟล์ที่เป็นไคร์เวอร์กราฟฟิก การกำหนดเป็นชื่อหรือค่าคงที่ก็ได้ ดังตารางที่ 8 ใช้ชื่อเป็น DETECT จะทำการตรวจสอบชนิดของวงจรที่แสดงผลให้เองพร้อมกับจะโหลดไฟล์ไคร์เวอร์กราฟฟิกสำหรับวงจรมานั้นเข้ามาให้องค์โดยอัตโนมัติ และโหมดเป็นการกำหนดขนาดของความละเอียดในการแสดงผลของแต่ละไคร์เวอร์ ซึ่งเราจะสามารถกำหนดเป็นชื่อหรือค่าคงที่ก็ได้ ดังตารางที่ 9

path คือการกำหนดเส้นทางเพื่อค้นหาไฟล์ไคร์เวอร์(ไฟล์ที่มีนามสกุลเป็น .BIG) โดยเขียนอยู่ภายในเครื่องหมายคำพูด ถ้าไม่กำหนดถือว่าไฟล์ไคร์เวอร์นั้นอยู่ในไดเรกทอรีปัจจุบัน ในการเข้าสู่โหมดกราฟฟิกโดยวงจรแสดงผลที่ใช้เป็นแบบ Hercules Graphics และมีความละเอียดขนาด 720 * 348 จุด และมีไฟล์ไคร์เวอร์อยู่ในไดเรกทอรีปัจจุบัน

```
#include<graphics.h>
main()
{
int driver,mode;
driver = HERC;
mode = HERCMONOH;
initgraph (&drive,&mode);
```

ตารางที่ 8
แสดงชื่อและค่าคงที่ของไครเวอร์ต่าง ๆ

ชื่อ	ค่าคงที่
DETECT	0
CGA	1
MCGA	2
EGA	3
EGA64	4
EGEMONO	5
OBM8514	6
HERCMONO	7
ATT400	8
VGA	9
PC3270	10

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 9

แสดงไครเวอร์ โหมด ค่าคงที่ และความละเอียดของระบบกราฟฟิก

ไครเวอร์	โหมด	ค่าคงที่	ความละเอียด
CGA	CGAC 0	0	320*200
	CGSC 1	1	320*200
	CGAC 2	2	320*200
	CGAC 3	3	320*200
	CGAHI	4	640*200
MCGA	MCGAC 0	0	320*200
	MCGAC 1	1	320*200
	MCGAC 2	2	320*200
	MCGAC 3	3	320*200
	MCGAMED	4	640*200
	MCGAHI	5	640*200
EGA	EGALO	0	640*200
	EGAHI	1	640*350
EGA64	EGA64LO	0	640*200
	EGAHI	1	640*350
EGAMONO	EGAMONOHI	3	640*350
	HERCMONOHI	0	720*348
	ATT400C 0	0	320*200
	ATT400C 1	1	320*200
	ATT400C 2	2	320*200
	ATT400C 3Q	3	320*200
	ATT400 CMED	4	640*200
	ATT400 CHI	5	640*400
AVGALO	VGALO	0	640*200
	VGAMED	1	640*350
	VGAHI	2	640*480
PC3270	PC3270HI	0	720*350
IBMB514	IBM8514LO	0	640*480

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การเข้าสู่โหมดกราฟฟิกอย่างอัตโนมัติและให้มีภาวะเป็น 0 พร้อมกับมีการกำหนดเส้นทางค้นหาไฟล์ ไดรเวอร์อยู่ที่ไดเรกทอรี C ภายใต้ไดเรกทอรีย่อย graph

```
#include<graphics.h>
main ( )
{
int gd = DETECT ,gm = 0;
initrph (&gd,&gm, "c:\graph");
```

3.6 ฟังก์ชันที่ใช้ในการเขียนโปรแกรม

ฟังก์ชันที่ใช้ในการเขียนโปรแกรม จะเป็นฟังก์ชันที่อยู่ในไฟล์ graphics.h ซึ่งผู้เขียนได้ใช้ในการเขียนโปรแกรม ดังรายละเอียดต่อไปนี้

3.5.1 circle เป็นฟังก์ชันที่ใช้ในการวาดรูปวงกลม ซึ่งมีรูปแบบเป็น

```
circle (x,y,radius);
```

x,y เป็นตำแหน่งของจุดศูนย์กลาง

radius เป็นรัศมีของวงกลมเป็นจำนวนจุด

3.5.2 cleardevice เป็นฟังก์ชันที่ใช้ลบหน้าจอโดยจะเหลือแต่สีพื้นที่ระบุไว้ในฟังก์ชัน setbkcolor สีพื้นที่จะเป็นสีค่าหรือสีของจอภาพ(คล้ายกับฟังก์ชัน clrscr ในเท็กซ์โหมด) ซึ่งมีรูปแบบเป็น

```
cleardevice ( );
```

3.5.3 closegraph เป็นฟังก์ชันที่ใช้ในการออกจากโหมดกราฟฟิกเท็กซ์โหมด ซึ่งมีรูปแบบเป็น

```
closegraph ( );
```

3.5.4 floodfill เป็นฟังก์ชันที่ใช้ระบายสีภายในรูปที่วาดขึ้นโดยรูปแบบและสีเป็นไปตามฟังก์ชัน setfillstyle หรือ setfillpattern ที่ใช้ก่อนหน้า ซึ่งมีรูปแบบเป็น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
floodfill (x,y,bordercolor);
```

x, y เป็นการกำหนดจุดที่อยู่ภายในรูปที่ต้องการระบายสี ถ้ากำหนดจุดอยู่นอกรูปจะระบายสีรอบนอกทั้งหมด ยกเว้นบริเวณภายในรูป
bordercolor เป็นขอบเขตของสีที่ต้องการให้ระบาย

3.5.5 `initgraph` เป็นฟังก์ชันเริ่มต้นการทำงานในโหมดกราฟฟิกโดยเปลี่ยนการทำงานจากเท็กซ์โหมดเป็นโหมดกราฟฟิก ซึ่งมีรูปแบบเป็น

```
initgraph (*graphdriver,*graphmode,*path);
```

3.5.6 `moveto` เป็นฟังก์ชันที่ใช้ในการเลื่อน cursor ไปยังตำแหน่งที่กำหนด มีรูปแบบเป็น

```
moveto (x,y);
```

3.5.7 `outtextxy` เป็นฟังก์ชันที่ใช้เมื่อต้องการนำข้อเสนอข้อความมาแสดงบนจอภาพที่ตำแหน่งที่เกิดขึ้น มีรูปแบบเป็น

```
outtextxy (x,y,*textstring);
```

x,y เป็นจุดเริ่มต้นที่จะพิมพ์ข้อความ `textstring` เป็นข้อความหรือตัวแปรชุดที่ต้องการนำมาแสดงผลในกราฟฟิกโหมด ถ้าเป็นข้อความต้องเขียนอยู่ภายในเครื่องหมายคำพูด (“.....”)

3.5.8 `rectangle` เป็นฟังก์ชันที่ใช้สร้างรูปสี่เหลี่ยม มีรูปแบบเป็น

```
rectangle (left,top,right,bottom);
```

`left, top` เป็นการกำหนดจุดมุมซ้ายบนของรูปสี่เหลี่ยม โดย `left` เป็นตำแหน่งคอลัมน์ `top` เป็นตำแหน่งแถว

right , bottom เป็นการกำหนดจุดมุมขวาด้านล่างของรูปสามเหลี่ยมโดย right เป็นตำแหน่งคอลัมน์ bottom เป็นตำแหน่งแถว

3.5.9 setcolor เป็นฟังก์ชันที่ใช้กำหนดสีที่จะใช้ในการวาดรูป มีรูปแบบเป็น

setcolor (color);\

color เป็นสีที่ต้องใช้วาดรูป โดยมีสีต่าง ๆ ดังตารางที่ 10

ตารางที่ 10
แสดงสีต่าง ๆ ที่มีในเทอร์โบซี

ชื่อ	ค่าคงที่	สี
BLACK	0	ดำ
BLUE	1	น้ำเงิน
GREEN	2	เขียว
CYAN	3	คราม
RED	4	แดง
MAGENTA	5	ม่วง
BROWN	6	น้ำตาล
LIGHTGRAY	7	เทาอ่อน
DARKGRAY	8	เทาเข้ม
LIGHTBLUE	9	น้ำเงินอ่อน
LIGHTGREEN	10	เขียวอ่อน
LIGHTCYAN	11	ครามอ่อน
LIGHTRED	12	แดงอ่อน
LIGHTMAGENTA	13	ม่วงอ่อน
YELLOW	14	เหลือง
WHITE	15	ขาว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.5.10 `setfillstyle` เป็นฟังก์ชันที่ใช้กำหนดรูปแบบ ของการระบายสีในบริเวณที่ต้องการ มีรูปแบบเป็น

`setfillstyle (pattern,color);`

3.5.11 `setlinestyle` เป็นฟังก์ชันที่ใช้กำหนดลักษณะของเส้นที่ใช้วาดภาพ มีรูปแบบเป็น

`setlinestyle (style,pattern,thickness);`

`ssyle` เป็นลักษณะของเส้นซึ่งสามารถกำหนดเป็นชื่อ หรือค่าคงที่ก็ได้ซึ่งแสดงได้

ดังตารางที่ 11

ตารางที่ 11

แสดงลักษณะของเส้นที่มีใช้ในเทอร์โบซี

ชื่อ	ค่าคงที่	ความหมาย
SOLID_LINE	0	เส้นทึบ
DOTTED_LINE	1	เส้นจุดไขว่ปลา
CENTER_LINE	2	เส้นผ่าศูนย์กลาง
DASHED_LINE	3	เส้นขีดกลาง
USERBIT_LINE	4	เส้นที่ผู้ใช้กำหนดเอง

`pattern` เป็นแบบของเส้น ซึ่งใช้เฉพาะกรณีที่กำหนด `style` ให้เป็น `USERBIT_LINE` `pattern` เป็นพารามิเตอร์ขนาด 16 บิต ตำแหน่งบิตที่มีค่าเป็น 1 จะเป็นจุดสว่าง ส่วนตำแหน่งบิตที่มีค่าเป็น 0 จะไม่มีจุดสว่าง

`thickness` เป็นการกำหนดความหนาของเส้น ซึ่งสามารถกำหนดเป็นชื่อหรือค่าคงที่ก็ได้ ดังตารางที่ 12

ตารางที่ 12

แสดงขนาดความกว้างของเส้น

ชื่อ	ครั้งที่	ความหมาย
NORM_WIDTH	1	เส้นหนาปกติ (กว้าง 1 พิกเซล)
THICK_WIDTH	2	เส้นหนา 3 เท่า (กว้าง 3 พิกเซล)

โดยฟังก์ชัน `setlinestyle` จะมีผลต่อฟังก์ชันที่กล่าวมาดังต่อไปนี้

`arc` `circle` `ellipse`
`line` `lineto` `rectangle`

3.7 คำสั่งควบคุมโปรแกรมภาษาซี

โดยปกติการทำงานของคอมพิวเตอร์ จะทำงานเรียงลำดับคำสั่งลงมาตั้งแต่ต้น โปรแกรมจนจบโปรแกรม แต่ถ้าเราต้องการเปลี่ยนแปลงขั้นตอนการทำงานของคำสั่ง เช่น โค้ดข้ามไปทำคำสั่งใดคำสั่งหนึ่ง หรือให้วนกลับมาทำคำสั่งที่เคยทำไปแล้วอีก ลักษณะการสั่งงานที่เป็นแบบนี้จะต้องใช้คำสั่งควบคุม ดังนั้นคำสั่งควบคุม จึงเป็นคำสั่งที่ใช้เปลี่ยนแปลงลำดับขั้นตอนการทำงานของโปรแกรม ซึ่งแบ่งออกได้เป็น 3 ชนิดคือ

3.6.1 คำสั่งให้ไปทำงาน โดยมีเงื่อนไข (conditional Branch Statement) ได้แก่

- คำสั่ง `if` เป็นรูปแบบคำสั่งในภาษาซีสามารถเขียนได้ 2 แบบดังนี้

```

if (condition)
{
    statement;
:
:
:
}

```

หรือ

```
if (condition)
    statement;
```

หาก condition เป็นจริงโปรแกรมจะทำคำสั่งในวงเล็บปีกกาซึ่งเป็นคำสั่งบล็อก แต่ถ้าหาก condition เป็นเท็จ โปรแกรมจะไม่ทำคำสั่งในบล็อก แต่ถ้าหาก statement เป็นคำสั่งเดียวไม่จำเป็นต้องใช้เครื่องหมายวงเล็บปีกกาก็ได้

- คำสั่ง if-else เป็นคำสั่งเงื่อนไขที่ทำงานเหมือนกับคำสั่ง if แต่มีโครงสร้างที่ง่ายต่อการทำความเข้าใจ และง่ายต่อการเขียนโปรแกรม รูปแบบของคำสั่ง if-else เป็นต้น

```
if (condition)
{
    statement_1;
    :
    /*block#1*/
    :
}
else
{
    statement_2;
    :
    /*block_2*/
    :
}
}
```

หรือ

```
if (condition)
    statement_1;
else
    statement 2;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดทั้งหมด อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

คำสั่ง if-else นั้นคล้ายกับคำสั่ง if ซึ่งโดยทั่วไปแล้วคำสั่งทั้งสองมักนำไปใช้งานที่เหมือนกัน ดังนั้นความแตกต่างของคำสั่งทั้งสองจึงอยู่ที่ว่าคำสั่ง if-else เป็นคำสั่งที่เป็นระเบียบมากกว่าคำสั่ง if และคำสั่ง if-else เป็นคำสั่งที่เข้าใจง่าย

การทำงานของคำสั่ง if-else เป็นดังนี้คือ ถ้าหาก condition เป็นจริงโปรแกรมจะทำคำสั่ง statement_1 หรือคำสั่งในบล็อก /*block#1*/ แต่ถ้าหาก condition เป็นเท็จ โปรแกรมจะไม่ทำคำสั่ง statement_1 หรือคำสั่งในบล็อก /* block#1 */ แต่จะทำคำสั่ง statement_2 หรือคำสั่งในบล็อก /*block#2*/ แทน

- คำสั่ง switch เป็นคำสั่งที่ใช้ตัวแปรเป็นเงื่อนไขและมีรูปแบบที่กะทัดรัดกว่าการใช้คำสั่ง if ดังรูปแบบของคำสั่งต่อไปนี้

```
switch (variable)
{
    case constant_1:
        statement;
    case constant_2:
        statement;
        :
        :
    default contant_3:
        statement;
}
```

รูปแบบของคำสั่ง switch เป็นการเลือกค่าตัวแปรหรือค่าคงที่โดยมีคำสั่ง case เป็นตัวนำไปสู่การทำงานของโปรแกรม ดังนั้นตัวเลือกของคำสั่ง switch ต้องเป็นตัวเดียวกับตัวเลือกของคำสั่ง case แต่ถ้าตัวเลือกของคำสั่ง switch ไม่ตรงกับตัวเลือกของคำสั่ง case เลข ดังนั้นโปรแกรมจะค้นหาคำสั่ง default ถ้าโปรแกรมพบคำสั่ง default โปรแกรมก็จะทำคำสั่งที่อยู่หลังคำสั่ง default แต่ถ้าโปรแกรมไม่พบคำสั่ง default โปรแกรมก็จะออกจากชุดคำสั่ง switch

3.6.2 คำสั่งให้ไปทำงานเป็นวงจหรือเป็นลูป (Loop Cotrol Statement) ได้แก่

- คำสั่ง `for` เป็นคำสั่งที่ต้องกำหนดเงื่อนไขถึง 3 ส่วน โดยส่วนแรกเป็นการกำหนดค่าเริ่มต้นของการนับส่วนที่ 2 เป็นการทดสอบเงื่อนไข และส่วนที่ 3 เป็นการกำหนดค่าการนับใหม่หรือการกำหนดการลดลงหรือเพิ่มขึ้น ดังรูปแบบคำสั่งทั่วไปต่อไปนี้

```
for (initialization; condition; increment)
    statement;
```

รูปแบบคำสั่งที่ใช้กับบล็อกของคำสั่งเป็นดังนี้

```
for (initialization; condition; increment)
{
    statement_1;
    statement_2;
    :
    :
    :
    statement_n;
}
```

โดยปกติแล้ว `initialization` ในคำสั่ง `for` เป็นตัวกำหนดค่าตัวแปรที่ใช้ควบคุมการทำงานของลูป ส่วน `condition` เป็นเทอมเงื่อนไขสำหรับการออกจากลูป และ สำหรับ `increment` เป็นส่วนที่กำหนดการเพิ่มหรือลดค่าตัวแปรที่ใช้ควบคุมการทำงานของลูป

- คำสั่ง `while` เป็นคำสั่งที่ต้องกำหนดเงื่อนไขซึ่งคล้ายกับคำสั่ง `for` โดยที่คำสั่งทั้งสองประกอบด้วยเงื่อนไข และชุดคำสั่งเหมือนกันแต่ต่างกันที่คำสั่ง `while` ไม่มีการกำหนดตัวแปรเริ่มต้นและไม่มีการกำหนดการเปลี่ยนแปลงตัวแปร สำหรับรูปแบบของคำสั่ง `while` เป็นดังนี้

```
while (condition) statement;
```

หรือ

```
while (condition)
```

```
{
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

statement_1;
statement_2;
:
:
statement_n;
}

```

รูปแบบของคำสั่ง while ทั้งสองแบบต่างกันที่คำสั่ง statement กล่าวคือ statement ในรูปแบบแรกเป็นคำสั่งบรรทัดเดียว ส่วน statement ในรูปแบบ 2 เป็นบล็อกของคำสั่งสำหรับ condition เป็นทอมเงื่อน ไขซึ่งถ้าหากค่าของ condition เป็นจริงหรือมีค่าที่ไม่เป็นศูนย์การทำงานก็คงอยู่ในรูป แต่ถ้า condition มีค่าเป็นเท็จการทำงานจะออกจากลูปทันที

- คำสั่ง do-while เป็นคำสั่งที่ไม่เหมือนกับคำสั่ง while หรือ คำสั่ง for โดยต่างกันที่ตำแหน่งการวางตัว condition สำหรับคำสั่ง do-while นั้นวาง condition ไว้ที่ตำแหน่งหลังสุดของคำสั่ง ซึ่งมีรูปแบบของคำสั่ง ดังนี้

```

do
{
statement_1;
statement_2;
:
:
statement_n;
} while (condition);

```

ถ้าหาก condition มีค่าเป็นจริงการทำงานก็คงอยู่ในรูป แต่ถ้า condition มีค่าเป็นเท็จการทำงานจะออกจากลูปสำหรับคำสั่ง do-while แตกต่างจากคำสั่ง while และคำสั่ง for ตรงที่ว่าถึงแม้ว่า condition เป็นเท็จ คำสั่งจะต้องทำในรอบแรกเสมอ

3.6.3 คำสั่งที่ทำไปโดยไม่มีเงื่อน ไข(Unconditional Branch Statement) ได้แก่

- คำสั่ง break เป็นคำสั่งที่ใช้ในลูปหรือร่วมกับคำสั่งลูปเพื่อหยุดการทำงานในลูปและออกจากลูป โดยจะทำงานต่อจากลูปจนหมดคำสั่ง สำหรับรูปแบบของคำสั่ง break

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

นั้นขึ้นกับคำสั่งดูอย่างไรก็แล้วแต่คำสั่ง `break` มักเป็นคำสั่งที่วางถัดจากคำสั่งเงื่อนไข ดังรูปแบบต่อไปนี้

รูปแบบในคำสั่ง `for`

```
for (initialization; condition; increment)
{
    statement;
    if (condition);
    break;
}
```

รูปแบบในคำสั่ง `while`

```
while (condition)
{
    statement;
    if (condition);
    break;
}
```

รูปแบบในคำสั่ง `do-while`

```
do
{
    statement;
    if (condition);
    break;
} while (condition);
```

คำสั่ง `break` สามารถใช้กับคำสั่งเงื่อนไขอื่น ๆ ที่นอกเหนือจากคำสั่ง `if` ซึ่งได้แก่ คำสั่ง `switch` โดยมีรูปแบบดังต่อไปนี้

รูปแบบคำสั่ง `switch`

```
switch (variable)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดตทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

{
    case constant_1:
        statement;
        break;
    case constant_2:
        statement;
        break;
    :
    :
    :
    default constant_3:
        statement;
}

```

คำสั่ง break ที่ใช้กับคำสั่ง switch มักเป็นคำสั่งที่วางถัดจากคำสั่งที่เป็น statement ซึ่งเป็น ส่วนของคำสั่ง case แต่ไม่ใช้กับคำสั่ง default

- คำสั่ง continue เป็นคำสั่งที่ทำงานตรงข้ามกับคำสั่ง break กล่าวคือ continue เป็นคำสั่งที่บังคับให้โปรแกรมกระโดดข้ามไปยังรอบต่อไป โดยคำสั่ง continue เป็นคำสั่ง ที่ใช้ร่วมกับคำสั่งเงื่อนไขซึ่งอยู่ระหว่างคำสั่ง loop โดยมีรูปแบบคำสั่งดังนี้

รูปแบบในคำสั่ง for

```

for (initialization; condition; increment)
{
    statement_1;
    if (condition);
    continue;
    statement 2;
}

```

รูปแบบในคำสั่ง while

```

while (condition)
{
    statement_1;
    if (condition);
    continue;
    statement_2;
}

```

รูปแบบในคำสั่ง do-while

```

do
{
    statement_1;
    if (condition);
    continue;
    statement_2;
} while (condition)

```

- คำสั่ง goto เป็นคำสั่งที่ใช้ในกลุ่มเช่นเดียวกับคำสั่ง break และคำสั่ง continue แต่สำหรับคำสั่ง goto ต้องใช้กับตัวเลขเสมอ ซึ่งตัวเลขนี้เป็นชื่อที่ถูกอ้างถึง โดยคำสั่ง goto นอกจากคำสั่ง goto สามารถใช้บังคับให้ออกจากกลุ่มแล้วคำสั่งนี้ยังสามารถทำหน้าที่เป็นคำสั่งลูปอีกด้วย รูปแบบของคำสั่งเป็นไปได้หลายแบบดังนี้

รูปแบบคำสั่งที่ทำหน้าที่เช่นเดียวกับคำสั่งลูป

```

label:
{
    statement_1;
    if (condition);
    goto label;
    statement_2;
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปแบบคำสั่งที่บังคับให้ออกจากลูปชั้นเดียว

```
for (initialization; condition; increment)
{
    statement_1;
    if (condition);
    goto label;
    statement_2;
}
label:
```

รูปแบบคำสั่ง goto แบบแรกนั้นเป็นคำสั่งที่ทำหน้าที่เช่นเดียวกับคำสั่งลูป ซึ่งถ้าหากว่า condition เป็นเท็จ โปรแกรมจะออกจากลูป ส่วนรูปแบบคำสั่ง goto แบบที่ 2 เป็นคำสั่งบังคับให้ออกจากลูปไปที่ตัวเลขที่ระบุในกรณีนี้ condition ต้องเป็นจริง โปรแกรมจึงจะออกจากลูป อย่างไรก็ตาม การใช้คำสั่ง goto ตามรูปแบบทั้ง 2 แบบนั้นสามารถใช้แทนได้ด้วยคำสั่ง for คู่กับคำสั่ง if หรือคำสั่ง while และหรือคำสั่ง do-while ซึ่งคำสั่งลูปเหล่านี้สามารถใช้ได้ดีและคล่องตัวกว่าคำสั่ง goto อย่างมาก

3.8 โปรแกรมสำหรับทดสอบ Hard Ware

สำหรับ Soft Ware ที่เขียนขึ้นเพื่อใช้ทดสอบ Hard Ware นี้จะใช้โปรแกรมหรือภาษาที่มีระดับสูงเพื่อความสะดวกและเข้าใจง่าย โดยงานวิจัยในวิทยานิพนธ์นี้จะใช้ภาษาซีเพื่อนำไปทดสอบชุด Hard Ware

3.7.1 โปรแกรมการทดสอบการ Interface

```
# include<stdio.h>
# include<dos.h>
main()
{
    int a=0 , b=0 , c=0 , d=0 , e=0;
    clrscr( );
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

printf ( "\ t [1] INPUT PORT\n");
printf ( "\ t [2] OUTPUT PORT\n");
printf ( "\ t SELECT VALUE 1 OR 2 :");
a=getchar ( );
getchar ( );
if (a ==1)

```

```
{
```

```

    outportd ( 771 , 155 );
    b=inport ( 768 );
    c=inport ( 769 );
    d=inport ( 770 );
    printf ( "\ t portA=%d \ tportB=%d \ tport C=%d\n" ,b,c,d );
}
else if ( a==2 )
{
    outportb ( 771 , 128 );
    for ( e==0 ; e<255 ; e++ )
    {
        outportb ( 768 , e );
        delay ( 200 );
        printf ( "%d \ t" , e );
    }
    for ( f==0 ; f<=255 ; f++ )
    {
        outport ( 769 , f );
        delay ( 200 );
        printf ( "%d \ t" , f );
    }
    for ( g==0 ; g<=255 ; g++ )

```

```

    {
        outportd ( 770, g );
        delay ( 200 );
        printf (“%d\t”, g );
    }
}
}

```

จากนั้นเชื่อมต่อชุดแปลงสัญญาณเข้ากับชุด Interface แล้วป้อนโปรแกรมทดสอบการ์ด Interface แล้วป้อนอินพุตซึ่งต่อกับคอนเนกเตอร์ IDC 16 โดยใช้ไฟ 0 V DC , 2.5 V, 5 V แล้วสังเกตผลจากจอภาพของเครื่อง PC ของค่านี้จะมีค่าเป็น 0, 128, 255 จากการป้อนอินพุต

3.7.2 โปรแกรมการทดสอบการ์ดทั้งหมด

```

#include<stdio.h>
#include<dos.h>
#define ONE ‘ 1 ’
#define TWO ‘ 2 ’
#define THREE ‘ 3 ’
main ()
{
    int a,b;
    clrscr ();
    printf (“\n\t[1] TEST SENCER SOUNE\n”);
    printf (“\t[2] TEST SENCER LIGHT\n”);
    printf (“\t[3] TEST SENCER THERMAL\n”);
    printf (“\t[1] SELECT VALU 1 OR 2 OR 3 :”);
    b = getchar ();
    outportd ( 771 , 144 );
    switch ( b )
    {
        case ONE :

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

getchar ();
outportb ( 770 , 00 );
for ( b=0 ; b<=200 ; b++)
{
a=inportb ( 768 );
delay ( 100 );
printf ( “5d\ t” , a );
}

getchar ();
break ;
case TWO :
getchar ();
break ;
case TWO :
getchar ();
outportb ( 770 , 03 );
for ( b=0 ; b<=200 ; b++)
{
a=inportb ( 768 );
delay ( 100 );
printf ( “%d\ t” , a );
}

getchar ();
break ;
case THREE :
getchar ();
outportb ( 770 , 06 );
for ( b = 0 ; b<= 200 ; b++)
{
a = inportb ( 768 );

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

delay ( 100 );
printf ( "%d \t", a );
}
getchar ( );
break ;
}
}

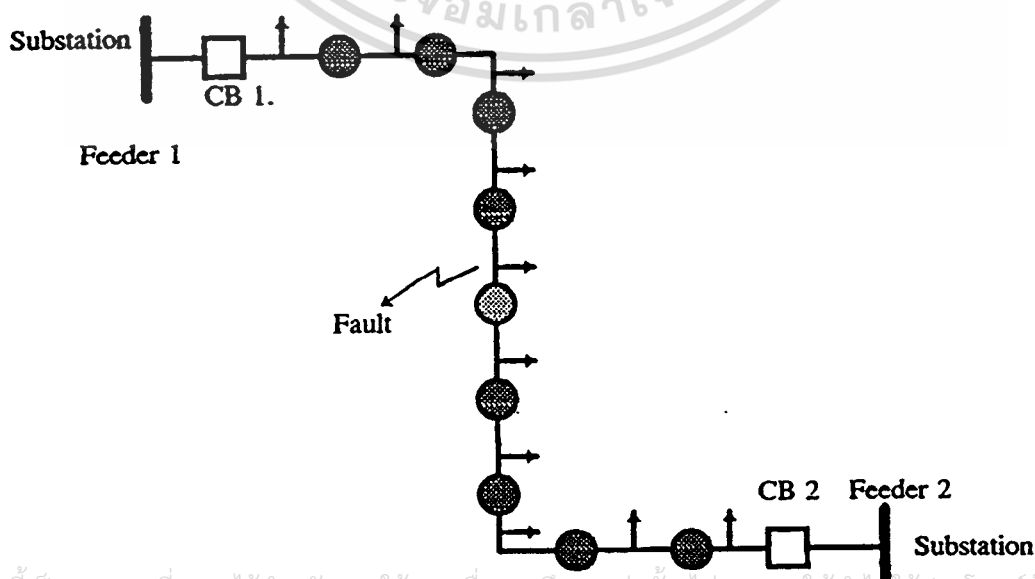
```

3.9 แบบจำลองการป้องกันและตรวจจับฟอลท์ในระบบจำหน่าย

การทำงานของระบบการส่งจ่ายกำลังไฟฟ้า ดังภาพที่ 43 จะเริ่มขึ้นเมื่อมีกระแสไฟฟ้าไหลเข้าหม้อแปลงไฟฟ้า ซึ่งเปรียบเสมือนสถานีไฟฟ้าย่อย(Substation)โดยทำการจ่ายกระแสไปยังผู้ใช้ไฟฟ้าตาม Node ต่าง ๆ ที่ต่ออยู่กับสายป้อน (Feeder) ของทั้ง 2 สถานีย่อย ซึ่งในสายป้อนนั้นยังมีอุปกรณ์ตัดต่อเพื่อแยกส่วนที่เกิดฟอลท์ที่เรียกว่า Automatic reclosing sectionlizer (Relay) ในสายป้อนของแต่ละสถานีออกซึ่ง Automatic reclosing sectionlizer หรือ Relay นี้ จะทำการติดตั้งในสายป้อนดังภาพที่ 44

ภาพที่ 43

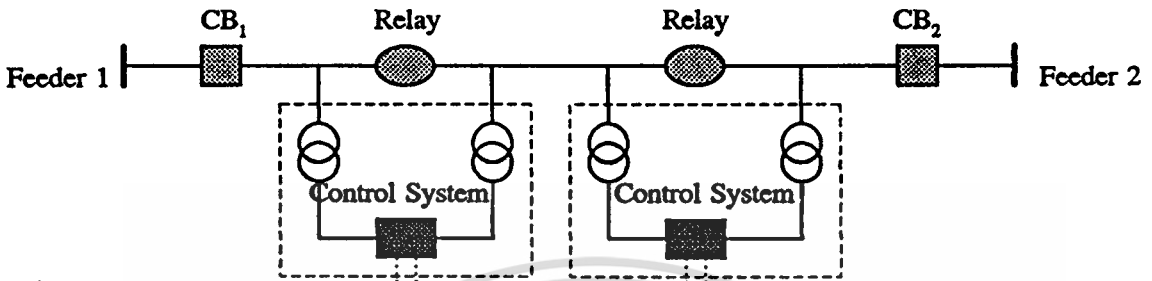
แสดง Single Line Diagram ของระบบส่งจ่ายกำลังไฟฟ้า



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาพที่ 44

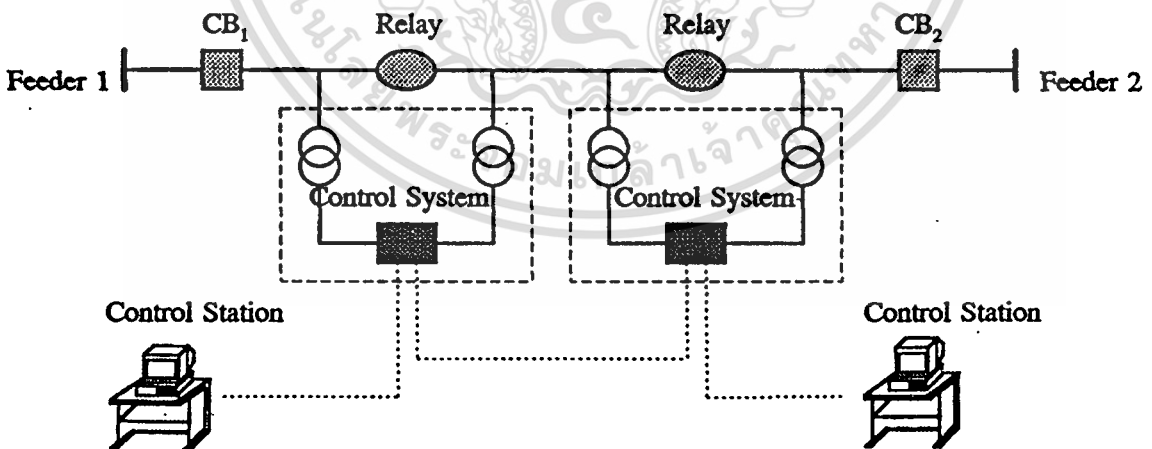
วิธีการติดตั้ง Automatic reclosing sectionlizer



จากภาพที่ 43 ระบบการส่งจ่ายกำลังไฟฟ้าจำลองนี้เป็นระบบไฟฟ้ากระแสตรง 12 โวลท์จ่ายไปยังผู้ใช้ไฟฟ้าตามจุด(Node)ต่าง ๆ โดยทั้ง 2 สถานีจ่ายไฟฟ้าข้อยนี้จะมีอุปกรณ์ที่เป็นจุด Tie line เพื่อทำการจ่ายไฟในส่วนที่ไม่เกิดฟอลต์ตามปกติ โดยใช้เวลาในการตรวจสอบส่วนที่เกิดฟอลต์ได้จากการส่งสัญญาณของตัวควบคุมมายังหน้าจอกอมพิวเตอร์ที่ศูนย์ควบคุม ดังภาพที่ 45

ภาพที่ 45

แสดงการทำงานระหว่างตัวควบคุมกับศูนย์ควบคุม



จากแบบจำลองระบบการส่งจ่ายนี้ สามารถแยกการเกิดฟอลต์ ได้ 2 ประเภท คือ Three Phase Fault และ Single Line To Ground Fault และจากฟอลต์ทั้ง 2 เมื่อเกิดขึ้นในระบบไฟฟ้าจะทำให้เกิดไฟฟ้าดับและเมื่ออุปกรณ์ป้องกัน Automatic reclosing sectionlizer (Relay) ตัดวงจรที่เกิดฟอลต์ออกไปและจ่ายไฟในส่วนที่ไม่เกิดฟอลต์ตามปกติ

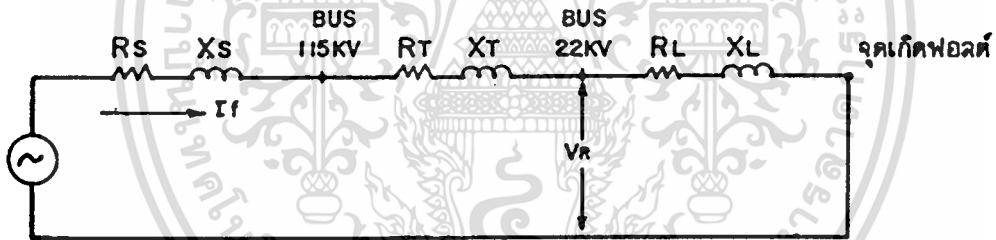
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.10 การหาค่ากระแสฟอลต์และแรงดันตกที่จุดต่าง ๆ ในระบบจำหน่าย

เมื่อเกิดฟอลต์ในระบบไฟฟ้าทั้ง Three phase fault และ Single line to ground ไม่เพียงแต่จะทำให้บริเวณที่เกิดฟอลต์เสียหายเท่านั้น แต่ยังทำให้กระแสลัดวงจร(Short circuit) ที่จ่ายมาจากแหล่งจ่ายไฟมายังจุดที่เกิดฟอลต์มีค่าสูงทั้งยังทำให้เกิดแรงดันตกผิดปกติ(Voltage Dip) ซึ่งช่วงเวลาที่เกิดแรงดันตกผิดปกตินี้จะขึ้นอยู่กับเวลาตั้งแต่เริ่มเกิดฟอลต์จนอุปกรณ์ป้องกัน(relay) ตัดส่วนที่เกิดฟอลต์ออกไปจากระบบ และถ้าแรงดันตกผิดปกตินี้มีมากกว่า 10 % ของแรงดันที่กำหนด จะส่งผลกระทบต่อการทำงานของอุปกรณ์ต่าง ๆ ได้

ภาพที่ 46

แสดง Single line diagram ของ Impedance ในระบบจากแหล่งจ่ายถึงจุดที่เกิดฟอลต์



พิจารณาจากภาพที่ 46 ค่าแรงดันในระบบ 22 kV. (V_R) เมื่อเกิดฟอลต์แบบ Three phase fault สามารถคำนวณหาได้จาก

$$\begin{aligned} V_R &= \frac{R_{1L} + jX_{1L}}{(R_{1S} + R_{1T} + R_{1L}) + j(X_{1S} + X_{1T} + X_{1L})} \times 22 \text{ kV.} \\ &= \frac{Z_{1L}}{Z_{1S} + Z_{1T} + Z_{1L}} \times 22 \text{ kV.} \end{aligned}$$

---(3-1)

และในกรณีที่เกิดฟอลต์แบบ Single line to ground fault ที่เฟส A หรือเฟสใดเฟสหนึ่ง สามารถคำนวณหาได้จากสมการ

$$V_{R(AN)} = \frac{(R_L + jx_L)}{(R_S + R_T + R_L) + j(x_S + x_T + x_L)} \times \frac{22 \text{ kV.}}{\sqrt{3}}$$

$$= \frac{Z_L}{Z_S + Z_T + Z_L} \times \frac{22 \text{ kV.}}{\sqrt{3}} \quad \text{---(3-2)}$$

และสามารถหาผลลัพท์ทางเวกเตอร์ของแรงดันได้เป็น

$$V_{R(AB)} = V_{R(AN)} - V_{R(BN)} \quad \text{---(3-3)}$$

$$V_{R(AC)} = V_{R(AN)} - V_{R(CN)} \quad \text{---(3-4)}$$

โดยที่ $V_{R(BN)} = V_{R(CN)} = \frac{22 \text{ kV.}}{\sqrt{3}}$ (เฟส B และ C ไม่เกิดฟอลต์)

เมื่อทราบค่าแรงดันที่ปรากฏในขณะที่เกิดฟอลต์ทั้ง 2 ประเภทแล้วนั้น และถ้าต้องการหาค่ากระแสลัดวงจรเมื่อเกิดฟอลต์ทั้ง 2 ประเภทสามารถหาได้จาก

$$I_{3-\phi} = \frac{V_{(PU)}}{Z_{1T(PU)}} \times I_{base} \quad \text{---(3-5)}$$

และ

$$I_{\phi-g} = \frac{3V_{(PU)}}{(Z_1 + Z_2 + Z_0)} \times I_{base} \quad \text{---(3-6)}$$

โดยที่ I_{base} = กระแสจากแหล่งจ่ายของระบบ 22 kV. , 100 MVA.

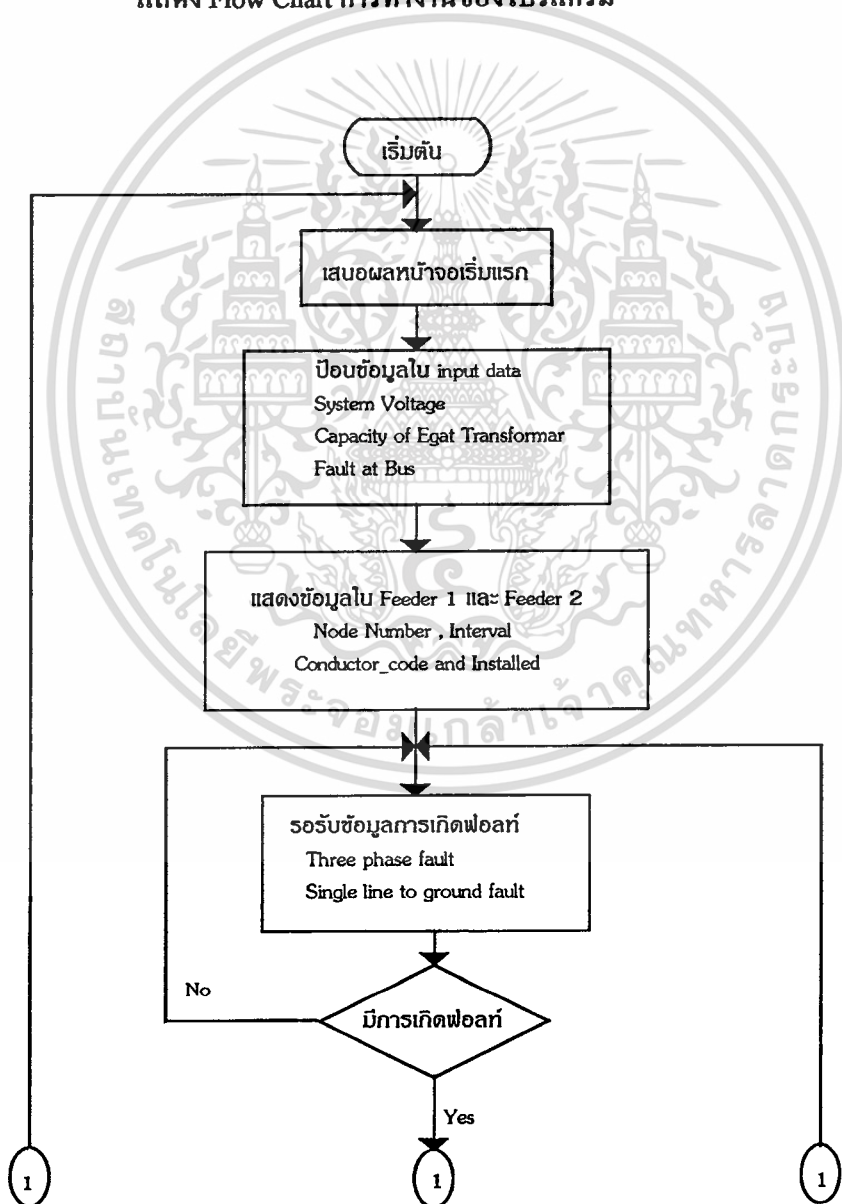
บทที่ 4

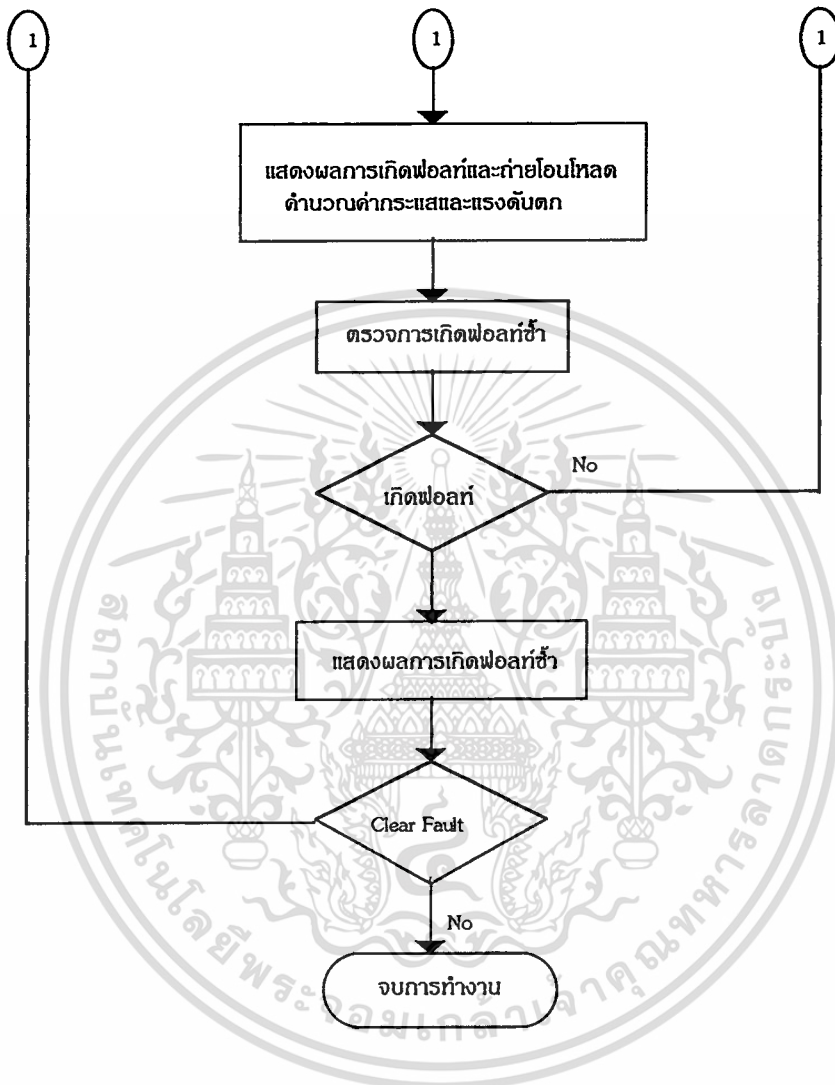
โปรแกรมและผลการตรวจจับฟอลต์ในระบบจำหน่าย

4.1 ขั้นตอนการใช้โปรแกรมร่วมกับชุดจำลอง

ภาพที่ 47

แสดง Flow Chart การทำงานของโปรแกรม





พิจารณา Flow Chart Diagram สามารถแสดงลำดับขั้นการใช้โปรแกรม ดังนี้

4.1.1 เริ่มต้นด้วยการแสดงผลที่หน้าจอคอมพิวเตอร์ด้วยภาพกราฟฟิก ดังภาพที่ 48

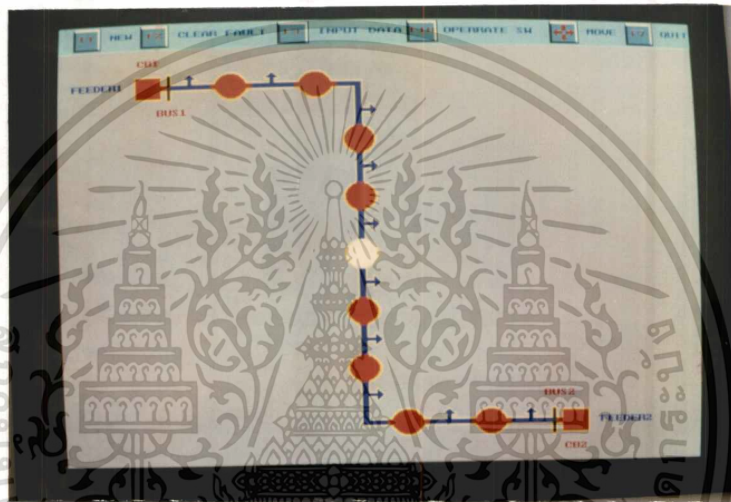
4.1.2 ผู้ควบคุมทำการป้อนข้อมูลลงใน Input Data ซึ่งใน Input Data นั้นประกอบด้วยข้อมูลของ System Voltage , Capacity of Egate Transformer และ Fault at Bus ดังภาพที่ 49

4.1.3 เมื่อผู้ควบคุมกำหนดข้อมูลใน Input Data เรียบร้อยแล้วผู้ควบคุมจำเป็นต้องกำหนดข้อมูลลงใน Data ของ Feeder 1 และ Feeder 2 ซึ่งประกอบด้วย Node Number , Interval , Conductor_Code และ Installed ดังภาพที่ 50 ถ้าต้องการ

ทราบว่า Conductor_Code ที่ใช้มีพื้นที่หน้าตัดของสายและชนิดของสาย
สามารถแสดงได้ดังภาพที่ 51

ภาพที่ 48

แสดงแผนผังของระบบจำหน่ายที่มี 2 Feeder



ภาพที่ 49

แสดงรูปแบบการป้อนข้อมูลลงใน Input Data

INPUT DATA

SYSTEM VOLTAGE : xx.xx (KV)

CAPACITY OF EGAT TRANSFORMER : xxx.xx (MVA)

FAULT AT BUS I-3 : xxxx.xx (A)

FAULT AT BUS I-g : xxxx.xx (A)

[Enter] to move [Esc] to quit

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาพที่ 50
แสดงรูปแบบการป้อนข้อมูลใน Feeder 1 และ Feeder 2

[FEEDER NO 1]				(Esc : to quit F1 : conductor code)
CB 1 CAPACITY (KVA): 5000				
NODE NO.	INTERVAL(Km)	CONDUCTOR_CODE	INSTALLED(KVA)	
1	5	5	15	
2	5	16	20	
3	5	5	25	
4	5	7	25	
5	5	10	30	

[FEEDER NO 1]				(Esc : to quit F1 : conductor code)
CB 1 CAPACITY (KVA): 5000				
NODE NO.	INTERVAL(Km)	CONDUCTOR_CODE	INSTALLED(kVA)	
1	5	5	15	
2	5	16	20	
3	5	5	25	
4	5	7	25	
5	5	10	30	

ภาพที่ 51
แสดงรูปแบบการค้นหา Conductor Code

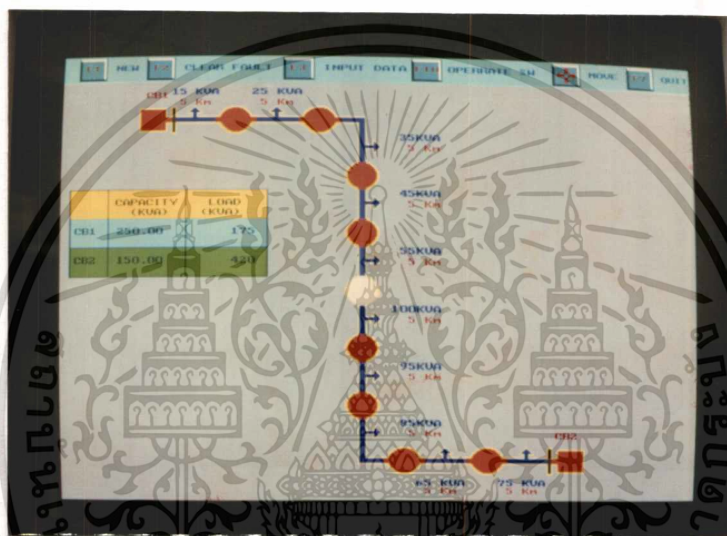
[FEEDER NO 1]				(Esc : to quit F1 : conductor code)
CB 1 CAPACITY (KVA): 5000				
NODE NO.	INTERVAL(Km)	CONDUCTOR_CODE	INSTALLED	CONDUCTOR : CODE
1	5	5	15	25A : 1
2	5	16	20	25A : 2
3	5	5	25	50A : 3
4	5	7	25	70A : 4
5	5	10	30	95A : 5
				120A : 6
				185 A : 7
				25AAY : 8
				50AAY : 9
				70AAY : 10
			25	95AAY : 11
1	5	16	20	25ACSR : 12
2	5	10	20	35ACSR : 13
3	5	10	35	50ACSR : 14
4	5	7	45	70ACSR : 15
5	5	10	25	95ACSR : 16

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากการป้อนข้อมูลลงใน Feeder 1 และ Feeder 2 เรียบร้อยแล้วข้อมูลที่กำหนดจะแสดงตามสถานที่ต่าง ๆ และแสดงผลเป็นภาพกราฟฟิกของ Single Line Diagram บนหน้าจอคอมพิวเตอร์ ดังภาพที่ 52

ภาพที่ 52

แสดงข้อมูลที่กำหนดลงใน Feeder 1 และ Feeder 2



และในส่วนของคุณจำลองที่สร้างขึ้น (Simulator Modelling) จะมีการจ่ายกระแสไฟฟ้าตามปกติ สามารถแสดงได้ดังภาพที่ 53

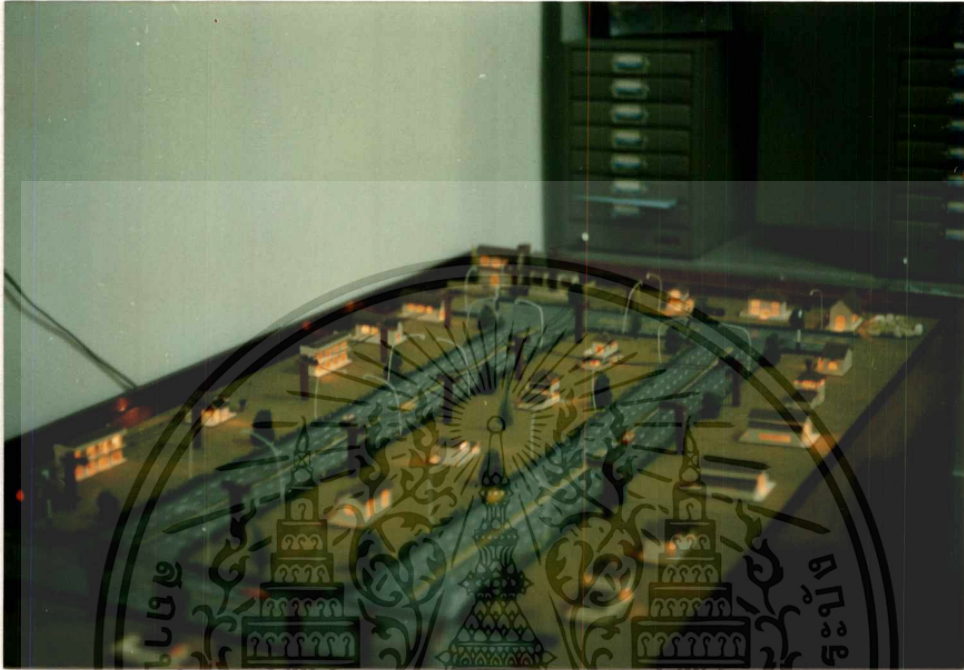
4.1.4 โปรแกรมจะรองรับข้อมูลการเกิดฟอลต์ โดยในงานวิจัยนี้ ได้ทำการจำลองการเกิดฟอลต์ออกเป็น 2 ประเภทคือ Three Phase Fault และ Single Line to Ground Fault ซึ่งสามารถแสดงการเกิดฟอลต์ได้ 3 กรณี ดังนี้คือ

- กรณีเกิด Three Phase Fault ขึ้นที่ Feeder 1 ในช่วง Node Number 2 ซึ่ง Automatic reclosing Sectionlizer หรือ Relay ที่ทำหน้าที่ปลด-สับ จะทำหน้าที่ตัดส่วนที่เกิดฟอลต์ที่ออกและส่งสัญญาณ โดยการแปลงสัญญาณอนาล็อกเป็นดิจิทัลและอินเตอร์เฟสกับ PC เพื่อแสดงผลการเกิดฟอลต์และโอนถ่ายโหลดจาก Feeder 1 ในส่วนที่ไม่เกิดฟอลต์ไปยัง Feeder 2 เพื่อให้มีกระแสไฟฟ้าใช้ตามปกติ และในขณะเดียวกันซอฟต์แวร์ที่เขียนขึ้นจะทำการประมวลผลเพื่อแสดงค่า Fault Current หรือ Short Circuit Current และ Voltage Drop at Bus ให้ผู้ควบคุมทราบได้ดังภาพที่ 54

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นอนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาพที่ 53

แสดงการจ่ายไฟในสภาวะปกติของทั้ง 2 Feeder จากชุดจำลองที่สร้างขึ้น (Simulator Modelling)



ภาพที่ 54

แสดงการเกิดฟอลต์แบบ Three Phase Fault ใน Feeder 1 ที่ Node Number 2

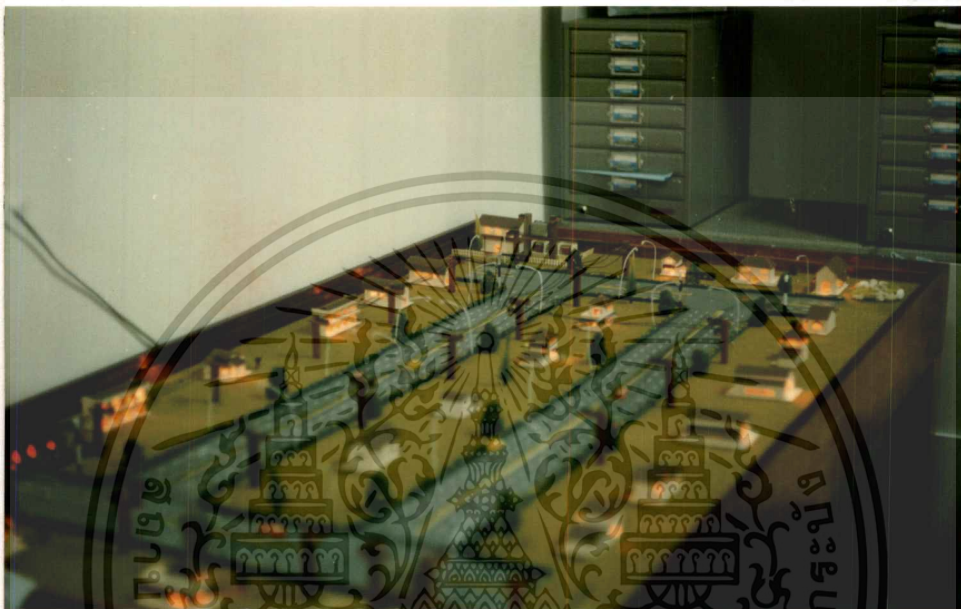


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

และในส่วนของชุดจำลองที่สร้างขึ้นจะเกิดบริเวณกระแสไฟฟ้าดับในบางส่วน ดังภาพที่ 55

ภาพที่ 55

แสดงบริเวณกระแสไฟฟ้าขัดข้องเมื่อเกิด Three Phase Fault จากชุดจำลองที่สร้างขึ้น

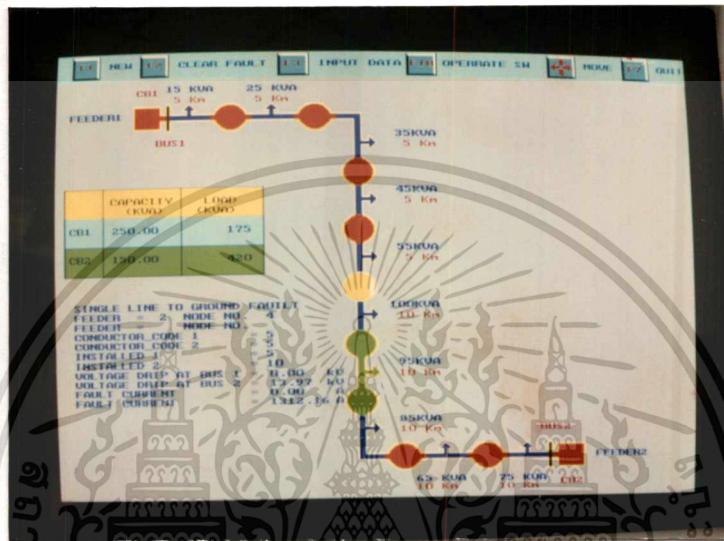


- กรณีเกิด Single Line to Ground Fault ขึ้นที่ Feeder 2 ในช่วง Node Number 4 ซึ่งอุปกรณ์ที่ทำหน้าที่ปลด-สับ จะทำหน้าที่ตัดในส่วนที่เกิดฟอลต์ออกและส่งสัญญาณโดยการแปลงสัญญาณอนาล็อกเป็นดิจิทัลและอินเตอร์เฟซกับ PC เช่นเดียวกับที่เกิด Three Phase Fault และแสดงผลการเกิดฟอลต์ขึ้นใน Feeder 2 ทางจอภาพคอมพิวเตอร์ โดยมีการโอนถ่ายโหลดเพื่อจ่ายไฟให้กับ Node Number 5 ให้มีกระแสไฟฟ้าใช้ตามปกติและในทำนองเดียวกัน ซอฟต์แวร์ที่เขียนขึ้นจะทำการประมวลผลเพื่อแสดงข้อมูลให้ผู้ควบคุมทราบได้ดังภาพที่ 56
- กรณีเกิด Single Line to Ground Fault และ Three Phase Fault พร้อมกัน โดยเกิด Single Line to Ground Fault ขึ้นที่ Feeder 2 ใน Node Number 4 และเกิด Three Phase Fault ขึ้นที่ Feeder 1 ใน Node Number 2 ที่ทำหน้าที่ปลด-สับ ตัดส่วนที่เกิดฟอลต์ออกและส่งสัญญาณแสดงผลการเกิดฟอลต์ออกทางจอภาพของคอมพิวเตอร์ โดยไม่สามารถโอนถ่ายโหลดให้กับ Node Number 5 ของ Feeder 2 ได้เนื่องจากเกิดบริเวณกระแสไฟฟ้าดับใน Feeder 1 จึงทำให้เกิดบริเวณกระแสไฟฟ้าดับเป็นบริเวณกว้าง และในส่วนของซอฟต์แวร์ก็จะทำการประมวลผลให้ผู้ควบคุมทราบเช่นกัน สามารถแสดงได้ดังภาพที่ 57

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

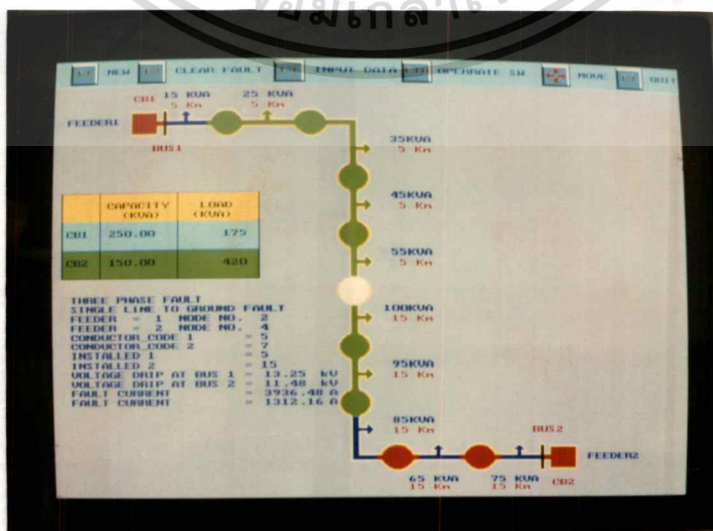
ภาพที่ 56

แสดงการเกิด Single Line to Ground Fault ใน Feeder 2 ที่ Node Number 4



ภาพที่ 57

แสดงการเกิด Single Line to Ground Fault และ Three Phase Fault ขันพร้อมกัน

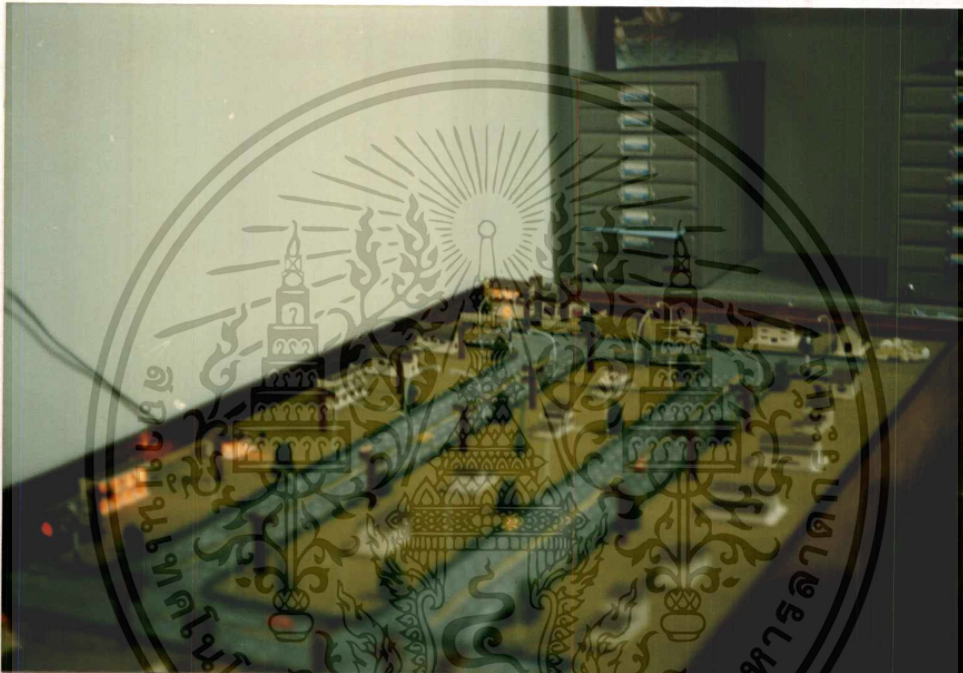


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

พิจารณาในส่วนของชุดจำลองที่สร้างขึ้น จะเห็นได้ว่าเกิดกระแสไฟฟ้าขัดข้องเป็นบริเวณกว้าง ดังภาพที่ 58

ภาพที่ 58

แสดงการเกิดกระแสไฟฟ้าขัดข้องทั้ง 2 Feeder จากชุดจำลองที่สร้างขึ้น



จากทั้ง 3 กรณี ดังกล่าว ซอฟต์แวร์ที่เขียนขึ้นจะมีการตรวจสอบการเกิดฟอลต์ใหม่อีกครั้ง โดยการหน่วงเวลาจากซอฟต์แวร์ซึ่งกำหนดไว้ 1 Sec และทำการแสดงผลซ้ำว่าเกิด ฟอลต์ในส่วนนั้นจริง จึงทำให้การโอนถ่ายโหลดพร้อมทั้งคำนวณค่ากระแสลัดวงจรและแรงดันตกชั่วขณะที่บัส

4.1.5 เมื่อทำการ โอนถ่ายโหลดและตัดส่วนที่เกิดฟอลต์ออกแล้ว ผู้ควบคุมจะสั่งการให้ตรวจสอบสาเหตุการเกิดฟอลต์และทำการ Clear Fault เพื่อให้ดำเนินการจ่ายไฟได้ตามปกติซึ่งเป็นการสิ้นสุดขั้นตอนการทำงานของ โปรแกรมที่ใช้ร่วมกันกับชุดจำลองที่สร้างขึ้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2 ผลการทดสอบจากโปรแกรม

จากซอฟต์แวร์และชุดจำลองที่สร้างขึ้น (Simulator Modelling) เป็นการทดสอบการเกิดฟอลต์ในระบบจำหน่ายพร้อมทั้งแสดงจุดที่เกิดฟอลต์และการปลด-สับสวิตช์โดยใช้ Relay เพื่อตัดส่วนที่มีฟอลต์ออกจากวงจรโดยแบ่งการเกิดฟอลต์เป็น 2 ประเภทที่เกิดขึ้นใน 2 Feeder ซึ่งจากซอฟต์แวร์ที่เขียนขึ้นสามารถประมวลผลเพื่อหาค่า Short Circuit Current และ Voltage Drip at Bus โดยการกำหนดขนาดของสายตัวนำโดยสามารถเลือก Conductor_Code เป็น 50 , 70 , 95 ,120 และ 185 Sq.mm. จะได้ผลการเกิด Short Circuit Current และ Voltage Drip at Bus ได้ดังนี้

- กรณีที่เกิด Three Phase Fault ขึ้นที่ Feeder 1 ในช่วง Node Number 2 ที่สายตัวนำขนาดต่างๆ สามารถประมวลผลได้ดังตารางที่ 13

ตารางที่ 13
แสดงผลการเกิด Three Phase Fault ขึ้นที่ Feeder 1 ที่ระยะทาง 5 km.

Feeder 1 ใน Node Number 2				
ขนาดสายตัวนำ (Sq.mm.)	Interval (km)	Short Circuit Current (A)	Voltage Drip at Bus (kV)	Percentage VoltageDrip (%)
50	5	2485.21	13.58	61.74
70	5	2780.45	13.13	59.72
95	5	3028.94	12.10	55.02
120	5	3160.88	11.46	52.13
185	5	3296.51	11.04	50.18

พิจารณาจากตารางที่ 13 เป็นการแสดงผลเมื่อมีการกำหนดระยะทางจาก Circuit Breaker 1 (CB₁) ถึง Node Number 2 เท่ากับ 5 km. และถ้าต้องการเปลี่ยนระยะทางโดยกำหนดระยะทางเป็น 10 , 15 , 20 และ 25 km. ซอฟต์แวร์สามารถประมวลผลได้ดังตารางที่ 14 , ตารางที่ 15 , ตารางที่ 16 และ ตารางที่ 17

ตารางที่ 14

แสดงผลการเกิด Three Phase Fault ชั้นที่ Feeder 1 ที่ระยะทาง 10 km.

Feeder 1 ใน Node Number 2				
ขนาดสายตัวนำ (Sq.mm.)	Interval (km)	Short Circuit Current (A)	Voltage Drip at Bus (kV)	Percentage Voltage Drip (%)
50	10	1757.80	16.79	67.35
70	10	1707.24	16.45	74.78
95	10	1930.48	15.61	70.98
120	10	2068.22	15.07	68.53
185	10	2206.51	14.77	67.14

ตารางที่ 15

แสดงผลการเกิด Three Phase Fault ชั้นที่ Feeder 1 ที่ระยะทาง 15 km.

Feeder 1 ใน Node Number 2				
ขนาดสายตัวนำ (Sq.mm.)	Interval (km)	Short Circuit Current (A)	Voltage Drip at Bus (kV)	Percentage Voltage Drip (%)
50	15	1025.49	18.23	82.88
70	15	1137.15	17.96	81.64
95	15	1410.02	17.29	78.58
120	15	1321.78	16.84	76.56
185	15	1760.23	16.25	73.86

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 16

แสดงผลการเกิด Three Phase Fault ขึ้นที่ Feeder 1 ที่ระยะทาง 20 km.

Feeder 1 ใน Node Number 2				
ขนาดสายตัวนำ (Sq.mm.)	Interval (km)	Short Circuit Current (A)	Voltage Drip at Bus (kV)	Percentage Voltage Drip (%)
50	20	789.84	19.04	86.59
70	20	951.79	18.82	85.57
95	20	1909.09	18.27	83.03
120	20	1210.71	17.89	81.33
185	20	1323.75	17.72	80.55

ตารางที่ 17

แสดงผลการเกิด Three Phase Fault ขึ้นที่ Feeder 1 ที่ระยะทาง 25 km.

Feeder 1 ใน Node Number 2				
ขนาดสายตัวนำ (Sq.mm.)	Interval (km)	Short Circuit Current (A)	Voltage Drip at Bus (kV)	Percentage Voltage Drip (%)
50	25	641.92	19.57	88.97
70	25	778.43	19.38	88.11
95	25	914.29	18.91	85.95
120	25	1002.15	18.58	84.48
185	25	1130.54	18.34	83.36

- กรณีที่เกิด Single Line to Ground Fault ขึ้นที่ Feeder 2 ในช่วง Node Number 4 ที่สายตัวนำขนาดต่าง ๆ และที่ระยะทางต่าง ๆ ซอฟต์แวร์สามารถประมวลผลได้ดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 18

แสดงผลการเกิด Single Line to Ground Fault ชั้นที่ Feeder 2 ที่ระยะทาง 5 km.

Feeder 2 ใน Node Number 4				
ขนาดสายตัวนำ (Sq.mm.)	Interval (km)	Short Circuit Current (A)	Voltage Drip at Bus (kV)	Percentage Voltage Drip (%)
50	5	1987.40	9.82	73.36
70	5	2157.53	9.39	73.97
95	5	2285.68	9.06	71.35
120	5	2356.62	9.07	71.45
185	5	2220.20	8.69	68.42

ตารางที่ 19

แสดงผลการเกิด Single Line to Ground Fault ชั้นที่ Feeder 1 ที่ระยะทาง 10 km.

Feeder 2 ใน Node Number 4				
ขนาดสายตัวนำ (Sq.mm.)	Interval (km)	Short Circuit Current (A)	Voltage Drip at Bus (kV)	Percentage Voltage Drip (%)
50	10	1078.42	10.14	79.90
70	10	1191.69	10.83	85.29
95	10	1347.63	10.59	83.45
120	10	1333.59	11.07	87.20
185	10	1349.19	11.33	81.33

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 20

แสดงผลการเกิด Single Line to Ground Fault ขึ้นที่ Feeder 2 ที่ระยะทาง 15 km.

Feeder 2 ใน Node Number 4				
ขนาดสายตัวนำ (Sq.mm.)	Interval (km)	Short Circuit Current (A)	Voltage Drip at Bus (kV)	Percentage Voltage Drip (%)
50	15	738.26	11.61	91.43
70	15	794.13	10.97	86.37
95	15	888.86	11.22	88.37
120	15	972.87	11.62	91.51
185	15	1020.12	12.59	92.26

ตารางที่ 21

แสดงผลการเกิด Single Line to Ground Fault ขึ้นที่ Feeder 1 ที่ระยะทาง 20 km.

Feeder 2 ใน Node Number 4				
ขนาดสายตัวนำ (Sq.mm.)	Interval (km)	Short Circuit Current (A)	Voltage Drip at Bus (kV)	Percentage Voltage Drip (%)
50	20	560.75	11.87	93.47
70	20	625.88	12.42	97.68
95	20	679.94	11.56	91.04
120	20	711.44	11.81	89.68
185	20	730.45	12.39	93.00

ตารางที่ 22

แสดงผลการเกิด Single Line to Ground Fault ขึ้นที่ Feeder 1 ที่ระยะทาง 25 km.

Feeder 2 ใน Node Number 4				
ขนาดสายตัวนำ (Sq.mm.)	Interval (km)	Short Circuit Current (A)	Voltage Drip at Bus (kV)	Percentage Voltage Drip (%)
50	25	541.95	12.02	94.64
70	25	550.55	12.48	89.27
95	25	550.60	11.78	92.75
120	25	576.69	11.91	93.78
185	25	622.50	11.97	94.30

พิจารณาจากตารางผลการทดสอบดังกล่าวข้างต้นสามารถเขียนกราฟแสดงความสัมพันธ์ต่าง ๆ ได้ดังนี้

- ความสัมพันธ์ระหว่างกระแสฟอลท์ (Short Circuit Current) กับระยะทาง (Interval)

$$I_F = f(I_N)$$

- ความสัมพันธ์ระหว่างเปอร์เซ็นต์ของแรงดันตกในขณะที่เกิดฟอลท์ (Percentage Voltage Dip) กับระยะทาง (Interval)

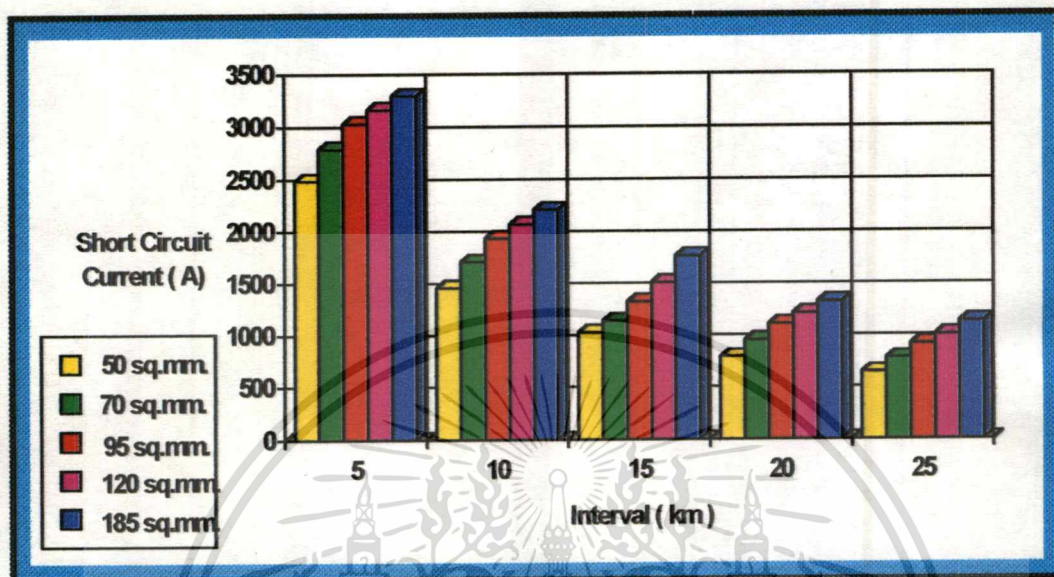
$$V_D = f(I_N)$$

- ความสัมพันธ์ระหว่างแรงดันตกที่บัส (Voltage Drip at Bus) กับขนาดของสายตัวนำ (Conductor_Code)

$$V_B = f(C_C)$$

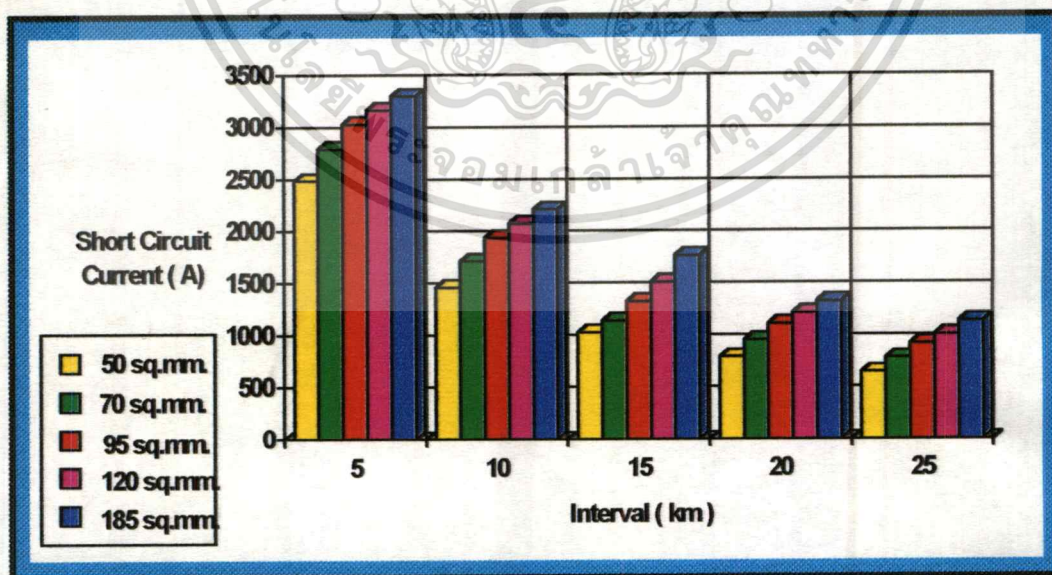
ภาพที่ 59

แสดงความสัมพันธ์ระหว่างกระแสฟอลท์(Short Circuit Current)กับระยะทาง(Interval) เมื่อเกิด Three Phase Fault ขึ้นที่ Feeder 1 ใน Node Number 2 ของสายตัวนำขนาดต่าง ๆ



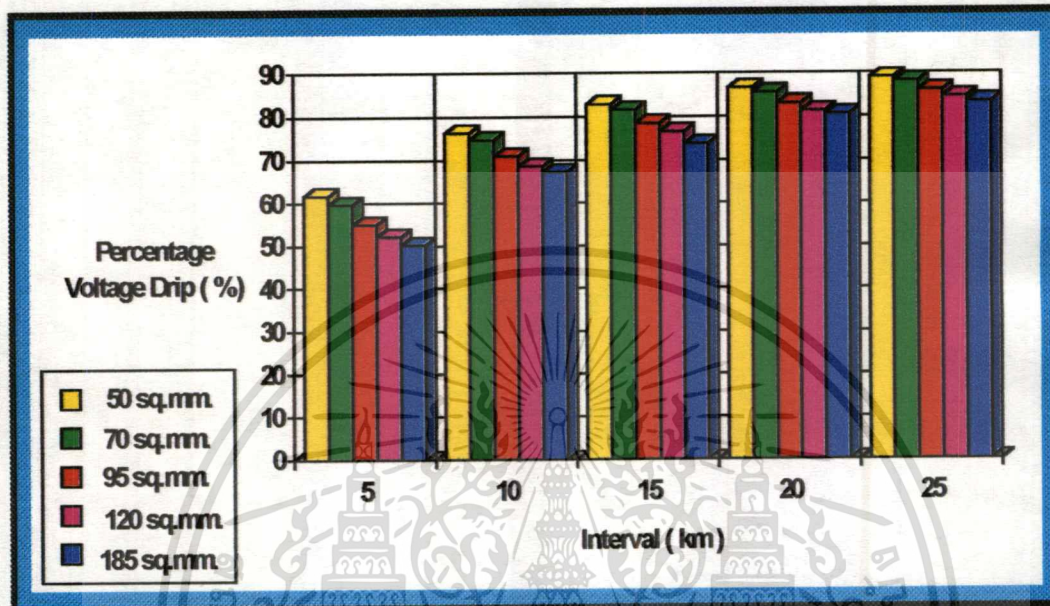
ภาพที่ 60

แสดงความสัมพันธ์ระหว่างกระแสฟอลท์(Short Circuit Current)กับระยะทาง(Interval) เมื่อเกิด Single Line to Ground Fault ขึ้นที่ Feeder 2 ใน Node Number 4 ของสายตัวนำขนาดต่าง ๆ



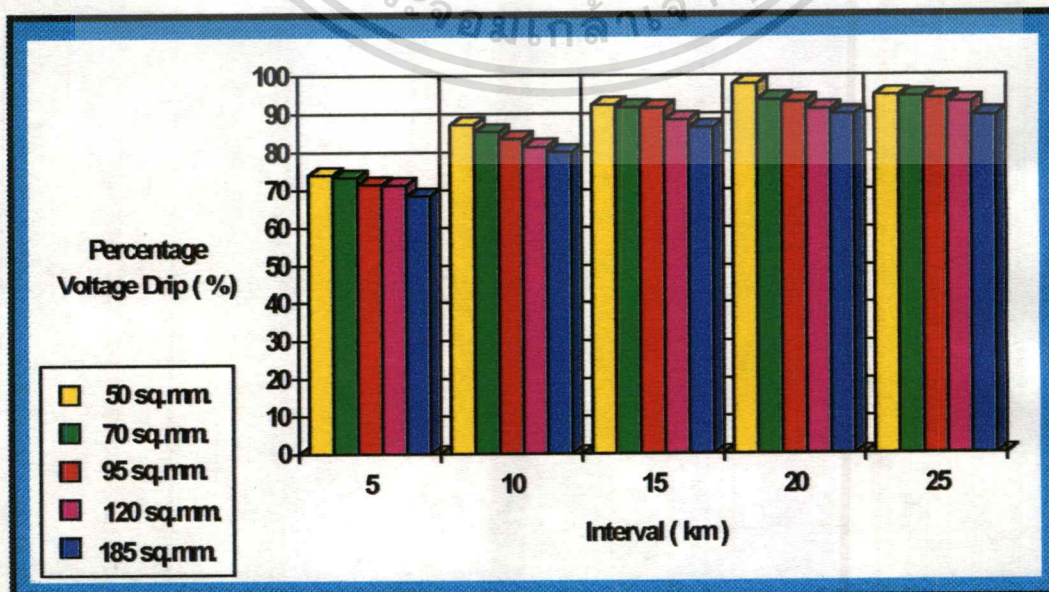
ภาพที่ 61

แสดงความสัมพันธ์ระหว่างแรงดันตก (Percentage Voltage Dip)กับระยะทาง(Interval) เมื่อเกิด Three Phase Fault ที่ Feeder 1 ใน Node Number 2 ของสายตัวนำขนาดต่าง ๆ



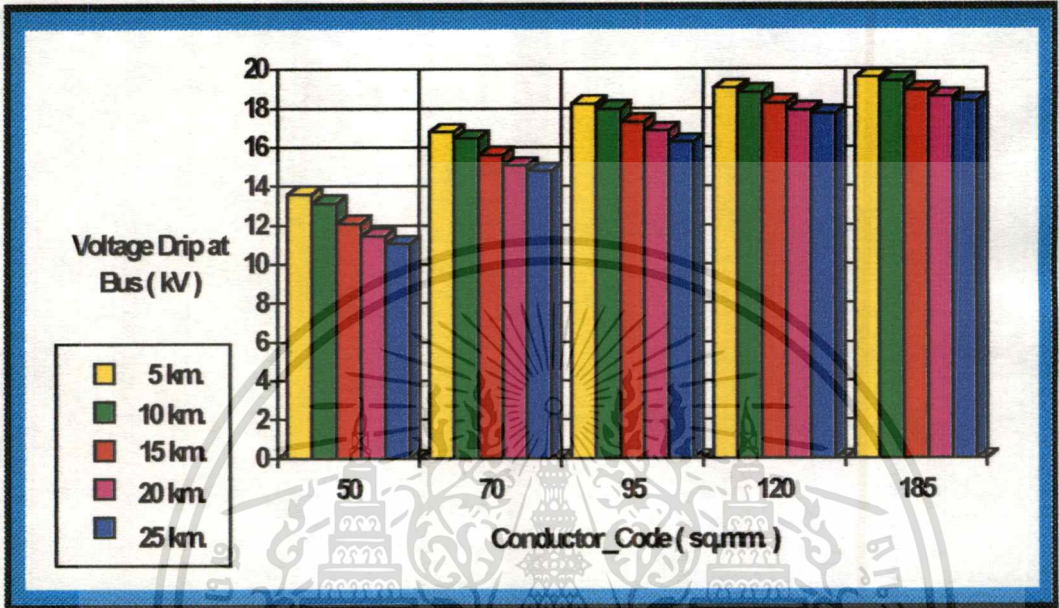
ภาพที่ 62

แสดงความสัมพันธ์ระหว่างแรงดันตก (Percentage Voltage Dip)กับระยะทาง(Interval) เมื่อเกิด Single Line to Ground Fault ที่ Feeder 2 ใน Node Number 4 ของสายตัวนำขนาดต่าง ๆ



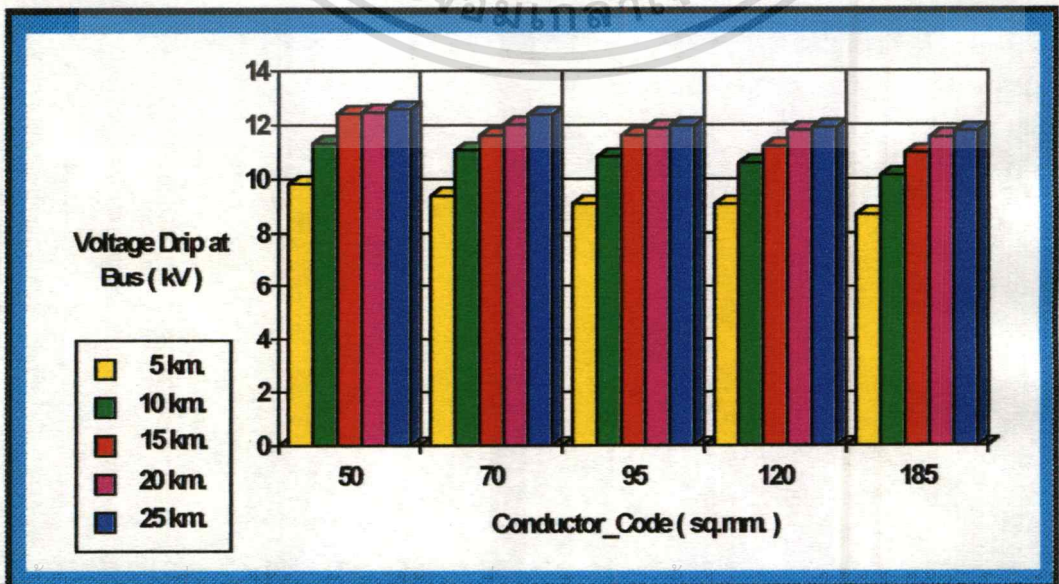
ภาพที่ 63

แสดงความสัมพันธ์ระหว่างแรงดันตกชั่วขณะที่บัส(Voltage Drip at Bus) กับขนาดของสายตัวนำ(sq.mm.)เมื่อเกิด Three Phase Fault ที่ Feeder 1 ใน Node Number 2



ภาพที่ 64

แสดงความสัมพันธ์ระหว่างแรงดันตกชั่วขณะที่บัส(Voltage Drip at Bus) กับขนาดของสายตัวนำ (Conductor_Code)เมื่อเกิด Single Line to Ground Fault ที่ Feeder 2 ใน Node Number 4



บทที่ 5

สรุปผลการวิจัยและข้อเสนอแนะ

5.1 สรุปผลการวิจัย

การศึกษาและจำลองการป้องกันในระบบจำหน่ายโดยอัตโนมัติ ที่นำเสนอในวิทยานิพนธ์นี้ ได้ศึกษาและนำเทคโนโลยีทางด้านคอมพิวเตอร์เข้ามาช่วยควบคุมการจ่ายกระแสไฟฟ้าในระบบจำหน่ายที่จำลองขึ้น โดยเขียนซอฟต์แวร์เพื่อควบคุมชุดจำลองและเนื่องจากซอฟต์แวร์นี้ได้ถูกพัฒนาให้สามารถติดต่อกับผู้ควบคุมในรูปแบบกราฟฟิก(Graphical)จึงเป็นผลดีที่ทำให้ผู้ควบคุมสามารถควบคุมการจ่ายกระแสไฟฟ้าในสายป้อนของระบบจำหน่ายได้ อีกทั้งยังสามารถพิจารณาพิภคของโหลดใน Node Number ต่าง ๆ ทั้ง 2 สายป้อน , ระยะทางที่ติดตั้งอุปกรณ์ป้องกันในสายป้อน , พิกัดของอุปกรณ์ป้องกันในสายป้อนและกระแสฟอลท์ที่เกิดขึ้นในระดับแรงดันต่าง ๆ และที่ขนาดสายตัวนำต่าง ๆ

จากที่กล่าวมาข้างต้นผู้วิจัยได้จัดสร้างชุดจำลองการป้องกันระบบจำหน่ายโดยอัตโนมัติ (simulator Modelling) เพื่อทำการทดสอบซอฟต์แวร์ที่เขียนขึ้น และเมื่อพิจารณาผลที่ได้จากตัวอย่างการทดสอบในบทที่ 4 ได้มีการทดสอบการเกิดฟอลท์ของทั้ง 2 สายป้อนโดยแบ่งชนิดของฟอลท์ออกเป็น 2 ประเภท คือ Single line to ground fault และ Three phase fault และเมื่อทำการจำลองการเกิดฟอลท์ขึ้น ผู้ควบคุมสามารถทราบถึงสถานะการทำงานของระบบและอุปกรณ์ที่ติดตั้งในสายป้อนด้วยสีต่าง ๆ เช่น

- สีแดง แสดงการจ่ายกระแสไฟฟ้าในสภาวะปกติ
- สีเขียว แสดงการเกิดกระแสไฟฟ้าขัดข้องในส่วนนั้น ๆ
- สีขาว แสดงอุปกรณ์ที่ทำหน้าที่โอนถ่ายโหลดจากแหล่งจ่ายหนึ่งสู่อีกแหล่งจ่ายหนึ่ง
- สีเหลือง แสดงสถานะของอุปกรณ์ที่ทำการ โอนถ่ายโหลดของอุปกรณ์จากแหล่ง

จ่ายหนึ่งสู่อีกแหล่งจ่ายหนึ่ง

เมื่อทราบสถานะการทำงานของอุปกรณ์ตัดตอนแล้วทำให้ผู้ควบคุมสามารถทราบสถานะที่เกิดฟอลท์และปริมาณของโหลดในช่วงที่เกิดฟอลท์นั้น ซึ่งซอฟต์แวร์ดังกล่าวจะเป็นประโยชน์อย่างมากต่อการวิเคราะห์และควบคุม นั่นคือ

5.1.1 ระยะเวลาในการค้นหาจุดที่เกิดฟอลท์ในระบบจำหน่าย

5.1.2 สามารถทราบข้อมูลต่าง ๆ ที่เกิดขึ้นในระบบจำหน่าย เช่น ปริมาณกระแสที่

เกิดฟอลท์แต่ละประเภท , แรงดันตกชั่วขณะที่บัส(Voltage Drip at Bus) เมื่อเกิดฟอลท์ ทำให้สามารถวิเคราะห์ข้อมูลดังกล่าวได้ว่า

- เมื่อเกิด Three Phase Fault จะทำให้แรงดันตกชั่วขณะที่บัสมากกว่า เกิด Single Line to Ground Fault ที่ระยะห่างจากจุดที่เกิดฟอลต์ถึงบัสที่เท่ากัน
- ถ้าระยะห่างจากจุดที่เกิดฟอลต์ถึงบัสมากขึ้น จะทำให้แรงดันตกชั่วขณะที่บัสลดลง
- ถ้าอิมพีแดนซ์จากแหล่งจ่าย(ฝ่ายผลิต)ถึงบัสลดลงแรงดันตกชั่วขณะจะลดลงตามไปด้วย
- ถ้าอิมพีแดนซ์ของสายตัวนำลดลงเนื่องจากการเปลี่ยนขนาดสายจะทำให้กระแสลัดวงจรมีค่ามากขึ้นทำให้แรงดันตกชั่วขณะที่บัสเพิ่มตามไปด้วย

5.1.3 สามารถควบคุมให้เกิดบริเวณ ไฟฟ้าดับน้อยที่สุดเมื่อพบฟอลต์โดยทำการโอนถ่ายโหลดจากสายป้อนหนึ่งสู่อีกสายป้อนหนึ่ง

5.1.4 สามารถนำซอฟต์แวร์ไปประยุกต์ใช้ในการเรียนการสอนในรายวิชาการป้องกันระบบได้

5.2 ข้อเสนอแนะ

ซอฟต์แวร์และชุดจำลองที่สร้างขึ้นนี้ เป็นการศึกษาและพัฒนาการป้องกันระบบจำหน่ายให้มีความสะดวกและรวดเร็วในการตรวจค้นหาจุดที่เกิดฟอลต์ได้ในระดับหนึ่ง ซึ่งอนาคตแล้วจำเป็นต้องมีการพัฒนาให้มีขีดความสามารถมากขึ้นต่อไป ซึ่งมีอยู่หลายแนวทางดังนี้

5.2.1 พัฒนาให้สามารถวิเคราะห์ฟอลต์ประเภทอื่น ๆ เพิ่มขึ้น เช่น Double Line to Ground Fault เป็นต้น

5.2.2 พัฒนาส่วนของการควบคุมให้สามารถควบคุมจาก Keyboard ไปสั่งการในระบบจำหน่ายได้

บรรณานุกรม

1. การไฟฟ้าส่วนภูมิภาค , เอกสารฝึกอบรมองค์การความร่วมมือระหว่างประเทศญี่ปุ่น
2. การไฟฟ้าส่วนภูมิภาค , ศูนย์ปฏิบัติการจ่ายไฟฟ้าย่อย นวนคร 1 และนวนคร 2
3. มন্ত্রী พจนารถลาวัณย์ , การเขียนโปรแกรมคอมพิวเตอร์ด้วยเทอร์โบซี , กรุงเทพฯ ฯ บริษัท ซีเอ็ดดูเคชั่น จำกัด (มหาชน) , 2537
4. ศรีณย์ อินทโกสุม , การเขียนโปรแกรมด้วยภาษาซี , กรุงเทพฯ ฯ สำนักพิมพ์แมคกรอ-ฮิล อินเทอร์เน็ตเนชั่นแนล เอ็นเตอร์ไพรส์, อิงค์ , 1996
5. A.C.M. Chen, "Automated Power Distribution" IEEE Spectrum, Vol.19, No.4, pp. 55-60, April 1982.
6. Aoki, Node Distribution line carrier System for section switch Telecontrol 1986 IEEE WM 229-9
7. Kato , Naito kohno , kanawa , Shoji Computer based Distribution automation IEEE , 1935 Power Industry computer Application Conference page 374-380
8. K.Kato, H.Nagasaka , A.Okimoto, M.Akizuki. " Development of Reliable Solidcore type Insulators and Current Limiting Arcing Horn for Distribution Lines" CEPSI Paper No.3.41 ,1986
9. K.Kato , H. Nagasaka, T.Ushida, T.Ohira. "Distribution Equipment for Hight reliability of Power Supply," CEPSI Paper No.3-31,1988
10. T. J. Kendrew , P. A. Schnieper, "Distributed Microprocessor Application to Electric Utility Distribution Protection and Control System", presented at the American control conference, San Diego, California, June 6-8, 1984.
11. R. Balakrishnan and A. Pahwa, "A Computer assisted intelligent storm outage evaluator for power distribution systems", IEEE Transactions on Power Delivery pp.1591-1597, July 1990.
12. R. D Christie, H. Zadehgo, and M. M. Habib, "High impedance fault detection in low voltage networks" , IEEE Transactions on Power Delivery, pp. 1829-1834, Oct 1993.
13. S. J. Balsler , K. A. Clements and D. J. Lawrence , " Amicroprocessor based technique for detection of high impedance faults" , IEEE Transactions on Power Delivery , pp. 252-258, July 1986.



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

/***** DAS. PROJECT *****/
/*****/
#include <stdio.h>
#include <graphics.h>
#include <bios.h>
#include <conio.h>
#include <dos.h>
#include <stdlib.h>
#include <math.h>
#include <string.h>
#define pic 20
#define xdim 40
#define ydim 30
#define row 10

struct sym_type {
    unsigned char image [pic][xdim][ydim];
}sym;

struct screen_type{
    unsigned int rec [170][1];
}screen;

struct feeder_status {
    unsigned int rec [33][5];
}feeder;

int f_data [14][3];

extern int recno = 10;
extern int run = 0;
extern int lcb1 =0;
extern int lcb2 = 0;

float system_v,trans_cap,fault_3,fault_g,cb1_cap,cb2_cap;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

unsigned int showf = 2;
unsigned int a;
main()
{
    int graphdriver = DETECT,graphmode;
    clrscr();
    initgraph(&graphdriver,&graphmode," ");
        simulation();
    closegraph();
}

/*****/
/* simulation */
/*****/
simulation()
{
    int x,y,ox,oy,size,xl,yl,mode;
    int ro = 10;
    int co =0;
    int chk = 0;
    int m_status;
    char tbuff[3];
    void *cursor;
    char ch;
    char gtext[40];
    x = 0 ; y = ydim*10 ;
    load_symbol();
    display_symbol(xdim*14,0,19,1); /*pic 19*/
    size = imagesize(xdim*14,0,(xdim*15)-1,30);

    cursor = malloc(size);

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

getimage(xdim*14,0,(xdim*15)-1,30,cursor);
mode = getgraphmode();
back_ground();
draw_picture ();
/* init_data (); /* install data */
/* feeder_data();
write_data();
check_load();*/
if(mouse_status() != 0) m_status = 1;
do {
    ch = 0 ;
    xl = yl = 0;
    /* putimage(x,y,cursor,XOR_PUT);*/
    ox = x ; oy = y ;
    chk = 0;
    while(chk != 1 ){
        if(m_status == 1){
            mouse_counters (&yl,&xl);
            if (xl != 0 || yl != 0)
                { if(xl > 0 ) ch = 77;
                  else
                    if(xl < 0 ) ch = 75;
                    if(yl > 0 ) ch = 80;
                    else
                      if(yl < 0 ) ch = 72;
                    chk =1;}
            if(mouse_press() == 2){

                while(mouse_release () !=0){
                    ch = 68;
                    chk = 1;}}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    }
    if (kbhit()){
        ch = getkey();
        chk = 1; }
    }
switch(ch) {
    case 75:/* left */
        x = x-xdim;
        recno=recno-row;co--;
        break;
    case 77:/* right */
        x = x+xdim;
        recno=recno+row;co++;
        break;
    case 72:/* up */
        y = y-ydim;
        recno=recno-1;ro--;
        break;
    case 80:/* down */
        y = y+ydim;
        recno=recno+1;ro++;
        break;
    case 59:/* F1: new */
        putimage(ox,oy,cursor,XOR_PUT);
        draw_picture();
        init_data();
        check_load();
        /*feeder_data();

        write_data();*/
        break;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

case 60:/* F2: clear fault */
    putimage(ox,oy,cursor,XOR_PUT);
    clear_fault();
/* showfault();*/
    break;
case 61:/* F3: input data */
    putimage(ox,oy,cursor,XOR_PUT);
    restorecrtmode();
    input_data();
    setgraphmode(mode);
    back_ground();
    drow_picture ();
    init_data ();
    write_data();
    check_load();
    output(0x307,0x0090);
loop: do{
    showf = inport(0x304);
    showfault();
    if (showf==127){setfillstyle(SOLID_FILL,7);
        bar(450,200,680,390);
        threepfault();}
    if (showf==239){setfillstyle(SOLID_FILL,7);
        bar(450,100,680,390);
        doublepfault();}
    if (showf==191){setfillstyle(SOLID_FILL,7);
        bar(450,100,680,190);
        setfillstyle(SOLID_FILL,7);
        bar(450,300,680,390);
        singlepfault();}
    if (showf==63){ setfillstyle(SOLID_FILL,7);

```

```

        bar(450,100,680,390);
        threepfault();singlepfault();}
if (showf==47){ setfillstyle(SOLID_FILL,7);
        bar(450,100,680,390);
        threepfault();singlepfault();doublepfault();}
if (showf==175){setfillstyle(SOLID_FILL,7);
        bar(450,100,680,390);
        doublepfault();singlepfault();}
if (showf==111){ setfillstyle(SOLID_FILL,7);
        bar(450,100,680,390);
        threepfault();doublepfault();}
if (showf==255){setfillstyle(SOLID_FILL,7);
        bar(450,100,680,390);}
}while (!kbhit());
ch=getkey();
switch (ch){
case 61:
        {goto loop;}
default:
        {closegraph();}
}
break;
case 67:
        /* putimage(ox,oy,cursor,XOR_PUT);*/
        show_data();
break;
case 68:/* F10: active sw. */
        putimage(ox,oy,cursor,XOR_PUT);
        run = screen.rec[recno][1];
        if(feeder.rec[run][1]==9||feeder.rec[run][1]==10||
        feeder.rec[run][1]==11||feeder.rec[run][1]==12||

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

feeder.rec[run][1]==4||feeder.rec[run][1]==5)
{ if(feeder.rec[run][3]==1)
    { if(feeder.rec[run][1]==9||feeder.rec[run][1]==4)
        putimage(ox,oy+30,cursor,XOR_PUT);
      else
        putimage(ox+40,oy,cursor,XOR_PUT);
      fault();
      tie_line();
    }
}
else if(feeder.rec[run][1] == 3)
    { if(feeder.rec[16][3]==1)
        tie_off();
      else
        tie_on();
    }
else if(feeder.rec[run][1]==1||feeder.rec[run][1]==2)
    { if(feeder.rec[ run ][3]==1)
        sw_off();
      else
        sw_on();
    }
break;
default:putimage(ox,oy,cursor,XOR_PUT);
}
if(x < 0) x = x+xdim;if(x > xdim*15) x = x-xdim;
if(y < 30)y = y+ydim;if(y > ydim*10) y = y-ydim;
if(ro < 1){ ro++;recno++;}
if(ro > 10){ ro--;recno--;}
if(co < 0){ co++;recno=recno+10;}
if(co > 15){ co--;recno=recno-10;}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        if(ch==75||ch==77||ch==72||ch==80||ch==71||ch==73||ch==79||ch==81)
            putimage(ox,oy,cursor,XOR_PUT);
    } while(ch != 65); /* F7      : quit */

free (cursor);
} /* end of control */
/*****/

show_data()
/*****/

{ int x=0,y=0,i,j;
  char buff[4];
  setfillstyle(1,7);
  bar(xdim,ydim*11,xdim*17,ydim*13);
  for(i=0;i<= 32;i++)
  { sprintf(buff,"%2d",i);
    setcolor(0);
    outtextxy(xdim+x,ydim*11,buff);
    x+=17;
  }
  for(i=3;i<= 5;i++)
  { x =0;
    for(j=0;j <= 32;j++)
    { sprintf(buff,"%2d",feeder.rec[j][i]);
      setcolor(0);
      outtextxy(xdim+x,ydim*11+15+y,buff);
      x+=17;
    }
    y+=15;
  }
}

/*****/

sw_off( )

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

/*****/
{ int count=0 ;
  device(feeder.rec[run][2],0);
  if(feeder.rec[run][4]==1)
    count = run+1;
  else
    count = run-1;
  do{
    if(feeder.rec[count][1]==9||feeder.rec[count][1]==10||
      feeder.rec[count][1]==11||feeder.rec[count][1]==12||
      feeder.rec[count][1]==4||feeder.rec[count][1]==5){
      device(feeder.rec[count][2],0);}
    if(feeder.rec[count][4]==1)
      count++;
    if(feeder.rec[count][4]==2)
      count--;
  }while(feeder.rec[count][3]!=0);
  check_load();
}
/*****/
sw_on()
/*****/
{ int count=0,stop=0,fd,ok=0 ;
  if(feeder.rec[run][4]==1 && feeder.rec[run-1][3]==1 )
    {fd = 1; }
  if(run == 0 ) {fd = 1; }
  if(feeder.rec[run][4]==2 && feeder.rec[run+1][3]==1 )
    {fd = 2; }
  if(run==32) {fd = 2;}
  if(fd == 1)
    {

```

```

if(feeder.rec[run-1][3]==1 && feeder.rec[run+1][3]==0 && feeder.rec
[run+1][5]==0 )
    {ok=1;}
if(feeder.rec[run][1]==1&&feeder.rec[run+1][3]==0 && feeder.rec[run+1]
[5]==0)
    {ok=1;}
    count = run+1;
    }
if(fd == 2)
    {
if(feeder.rec[run+1][3]==1&&feeder.rec[run-1][3]==0 && feeder.rec[run-1]
[5]==0)
    {ok=1;}
if(feeder.rec[run][1]==1&&feeder.rec[run-1][3]==0 && feeder.rec[run-1]
[5]==0)
    {ok=1;}
    count = run-1;
    }
if(ok ==1)
    { device(feeder.rec[run][2],fd);
    feeder.rec[run][4]=fd;
do{
    feeder.rec[count][4]=fd;
if(feeder.rec[count][1]==9||feeder.rec[count][1]==10||feeder.rec[count]
[1]==11||feeder.rec[count][1]==12
    ||feeder.rec[count][1]==4||feeder.rec[count][1]==5){
device(feeder.rec[count][2],fd);}
if(fd==1)
    count++;
if(fd==2)
    count--;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        if(feeder.rec[count][1]==2||feeder.rec[count][1]==3)
            if(feeder.rec[count][3]==0) stop =1;
        }while (stop!=1 );
    }
    check_load();
}
/*****/
fault( )
/*****/
{ int count = 0,fd,count_sw = 0,stop = 0 ;
  fault_sound();
  if(system_v != 0)
      calculate_fault();
  feeder.rec[run][5] = 1; /*get fault position*/
  if(run==13||run==14||run==15){ feeder.rec[13][5]=0;feeder.rec[14][5]=0;
    feeder.rec[15][5]=1;}
  if(run==17||run==18||run==19){ feeder.rec[17][5]=1;feeder.rec[18][5]=0;
    feeder.rec[19][5]=0;}
  if(feeder.rec [run][4]==1) /* set start for sw_off */
    { fd = 1;count = 0 ; run = 0;}
  if(feeder.rec [run][4]==2)
    { fd = 2;count = 32; run = 32;}
  sw_off(); /* 1st breaker trip */
  if(fd==1)count = 2 ; /* set start */
  if(fd==2)count = 31;
  do{
    if(feeder.rec[count][1]==2||feeder.rec[count][1]==3) /* all sw open */
      { device(feeder.rec[count][2],0); }
    if(fd==1)count++;
    if(fd==2)count--;
    if(feeder.rec[count][1]==2||feeder.rec[count][1]==3)

```

```

        if(feeder.rec[count][3]==0) stop=1;
    }while(stop != 1 );
    delay(700);
    if(fd==1)count = 0 ;           /* set start      */
    if(fd==2)count = 32;
    stop=0;                        /* 1st close breaker */
    do{                             /* close sw go to fault */
        if(fd == 1)
        { if(feeder.rec[count][2] == 133)
            { device(feeder.rec[count][2],fd);
              device(feeder.rec[count+1][2],fd);
              device(feeder.rec[count+2][2],fd);
              device(feeder.rec[count+3][2],fd);
              count+=4; delay(1000);stop = 1;
            }
          else
            { device(feeder.rec[count][2],fd);
              device(feeder.rec[count+1][2],fd);
              if(feeder.rec[count+1][5]==1)stop=1;
              count+=2;
            }
          count_sw++;
        }
        if(fd == 2)
        { if(feeder.rec[count][2] == 139)
            { device(feeder.rec[count][2],fd);
              device(feeder.rec[count-1][2],fd);
              device(feeder.rec[count-2][2],fd);
              device(feeder.rec[count-3][2],fd);
              count-=4; delay(1000);stop = 1;
            }
        }
    }

```

```

        else
        { device(feeder.rec[count][2],fd);
          device(feeder.rec[count-1][2],fd);
          if(feeder.rec[count-1][5]==1)stop=1;
          count-=2;
        }
        count_sw++;
    }
    check_load();
    delay(1000);
} while (stop !=1 ) ;
if(fd==1)count = 0 ;          /* set start */
if(fd==2)count = 32;
device(feeder.rec[count][2],0); /* 2nd breaker trip */
if(fd==1)count++ ;          /* set start */
if(fd==2)count--;
check_load();
/*line off*/
while (feeder.rec[count][5]!=1)
{ if(feeder.rec[count][1]==9||feeder.rec[count][1]==10||feeder.rec[count]
[1]==11||feeder.rec[count][1]==12
||feeder.rec[count][1]==4||feeder.rec[count][1]==5)
{ device(feeder.rec[count][2],0);}
  if(fd==1)count ++ ;
  if(fd==2)count -- ;
  check_load();
}
device(feeder.rec[count][2],0);
if(fd==1)count = count_sw*2-2 ;    /* set sw before fault */
if(fd==2)count = 32-(count_sw*2)+2;
device(feeder.rec[count][2],0);    /* open sw */

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

check_load();
    if(fd==1)count = 0 ;          /* set start */
    if(fd==2)count = 32;
    delay(1000);
    if( count_sw*2-2 != 0 || 32-(count_sw*2)+2 != 32)
        { device(feeder.rec[count][2],fd); } /* 2nd close breaker */
    check_load();
    if(fd==1)count++;           /* set start */
    if(fd==2)count-- ;
    stop = 0;
    if(count_sw*2-2 != 0||32-(count_sw*2)+2 !=32)
    { do{
        if(feeder.rec[count][1]==9||feeder.rec[count][1]==10||feeder.rec[count]
[1]==11||feeder.rec[count][1]==12
        ||feeder.rec[count][1]==4||feeder.rec[count][1]==5 )
        { device(feeder.rec[count][2],fd); }
        if(fd==1)count++ ;
        if(fd==2)count-- ;
        if(feeder.rec[count][1]==2||feeder.rec[count][1]==3)
            if(feeder.rec[count][3]==0) stop=1;
        check_load();
    }while(stop!=1);
    }
}
/*****/
clear_fault()
/*****/
{ int i,x,y,fault;
    for(i=0;i<=32;i++)
    {
        if(feeder.rec[i][5] == 1)

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

{ fault = feeder.rec[i][2];feeder.rec[i][5] = 0;
  if (fault==144||fault==145||fault==147||fault==148||
      fault==143||fault==149)
  {
    setfillstyle(1,7);
    bar(xdim*15,ydim*2,xdim*17,ydim*11);
    fault+=10;}
  else
    { fault++;}
  x = fault/10;
  y = fmod(fault,10); if (y == 0) y = 10;
  if (y == 10) x = x-1 ;
  x = xdim*x ; y = ydim*y;
  setfillstyle(1,7);
  bar(x,y,x+xdim,y+ydim-3);
  bar(xdim*7-8,ydim*5,xdim*13+8,ydim*7);
}
}
}
/*****/
tie_line()
/*****/

{ int i,fd=3,count,stop=0;
  if(feeder.rec[15][3] == 1) {fd = 1 ;count =17;} /* feeder 1 on */
  if(feeder.rec[17][3] == 1) {fd = 2 ;count =15;} /* feeder 2 on */
  if(feeder.rec[15][3]==1 && feeder.rec[17][3]==1) fd = 3;
  if( feeder.rec[17][5]==1 || feeder.rec[18][5]==1 ||feeder.rec[19][5]==1) fd = 3;
  if( feeder.rec[13][5]==1 || feeder.rec[14][5]==1 ||feeder.rec[15][5]==1) fd = 3;
  if(fd == 1 || fd ==2)
    { device(146,fd);delay(200); /* close tie sw. */

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

/*check_load()*/
feeder.rec[16][4] =fd; } /*set tie sw.feeder */
if (fd ==1)
{ stop=0;
do{
    if(feeder.rec[count+1][5]==1){stop=1;}
    if(stop == 0)
    { device(feeder.rec[count][2],fd);
      feeder.rec[count][4] = fd;
      delay(200);
      check_load();
      if(lcb1 > cb1_cap)
      { stop =1 ; overload(1);
        delay(1000);
        device(feeder.rec[count][2],0);
        feeder.rec[count][4] = 2;
        device(feeder.rec[count-1][2],0);
        feeder.rec[count-1][4] = 2;
        check_load();
      }
      count++ ;
    }
  }while(stop!=1);
}
if (fd ==2)
{ stop = 0;
do{
    if(feeder.rec[count-1][5]==1){stop=1;}
    if(stop==0)
    { device(feeder.rec[count][2],fd);

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

feeder.rec[count][4] = fd;
    delay(200);
    check_load();
    if(lcb2 > cb2_cap)
        { stop =1 ;
          delay(1000); overload(2);
          device(feeder.rec[count][2],0);
          feeder.rec[count][4] = 1;
          device(feeder.rec[count+1][2],0);
          feeder.rec[count+1][4] = 1;
          check_load();
        }
    count-- ;
}
}while(stop!=1);
}
}
/******/
tie_on()
/******/
{ int i,fd=3,count,stop=0;
  if(feeder.rec[15][3] == 1) fd = 1 ; /* feeder 1 on */
  if(feeder.rec[17][3] == 1) fd = 2 ; /* feeder 2 on */
  if(feeder.rec[15][3]==1 && feeder.rec[17][3]==1) fd = 3;
  if(feeder.rec[15][5]==1 || feeder.rec[17][5]==1) fd = 3;
  if(fd == 1 || fd ==2)
  { device(146,1); check_load();
    feeder.rec[16][4] =fd;} /*set feeder */
  stop = 0;
  if (fd ==1)
    { count = 17;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

do{
    if(feeder.rec[count][1]==9||feeder.rec[count][1]==10||
        feeder.rec[count][1]==11||feeder.rec[count][1]==12||
        feeder.rec[count][1]==4||feeder.rec[count][1]==5 )
    { device(feeder.rec[count][2],fd);feeder.rec[count][4] = fd;}
    count++ ;
    if(feeder.rec[count][1]==1||feeder.rec[count][1]==2||
        feeder.rec[count][1]==3 )
        { feeder.rec[count][4] = fd;
          if(feeder.rec[count][3]==0) stop=1; }
    check_load();
}while(stop!=1);
}
stop = 0 ;count--;
do{
    check_load();
    if(lcb1 <= cb1_cap)stop = 1;
    if(lcb1 > cb1_cap)
        { overload(1);
          delay(1000);
          device(feeder.rec[count-1][2],0);
          feeder.rec[count-1][4] = 2;
          device(feeder.rec[count][2],0);
          feeder.rec[count][4] = 2;
          count-=2;
          check_load();
        }
}while(stop != 1);
stop = 0 ;
if (fd ==2)
    { count=15;

```

```

do{
    if(feeder.rec[count][1]==9||feeder.rec[count][1]==10||
        feeder.rec[count][1]==11||feeder.rec[count][1]==12||
        feeder.rec[count][1]==4||feeder.rec[count][1]==5 )
    { device(feeder.rec[count][2],fd);feeder.rec[count][4] = fd; }
    count-- ;
    if(feeder.rec[count][1]==1||feeder.rec[count][1]==2||
        feeder.rec[count][1]==3 )
        { feeder.rec[count][4] = fd;
          if(feeder.rec[count][3]==0) stop=1; }
    check_load();
}while(stop!=1);
}
stop = 0 ;count++;
do{
    check_load();
    if(lcb2 <= cb2_cap)stop = 1;
    if(lcb2 > cb2_cap)
    { overload(2);
      delay(1000);
      device(feeder.rec[count+1][2],0);
      feeder.rec[count+1][4] = 1;
      device(feeder.rec[count][2],0);

      feeder.rec[count][4] = 1;
      count+=2;
      check_load();
    }
}while(stop != 1);
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

/*****/
tie_off()
/*****/

{ int i,fd=3,count,stop=0;
  if(feeder.rec[16][4] == 1) {fd = 1 ;count =17;} /* feeder 1 on */
  if(feeder.rec[16][4] == 2) {fd = 2 ;count =15;} /* feeder 2 on */
  if(fd == 1 || fd ==2)
    { device(146,0); check_load(); }

  stop = 0;
  if (fd ==1)
    { do{
      if(feeder.rec[count][1]==9||feeder.rec[count][1]==10||feeder.rec[count]
[1]==11||feeder.rec[count][1]==12||feeder.rec[count][1]==4||feeder.rec[count][1]==5)
        { device(feeder.rec[count][2],0);}
        count++ ;
        if(feeder.rec[count][1]==1||feeder.rec[count][1]==2||
feeder.rec[count][1]==3 )
          { if(feeder.rec[count][3]==0) stop=1; }
        check_load();
      }while(stop!=1);
    }
  stop = 0;
  if (fd ==2)
    { do{
      if(feeder.rec[count][1]==9||feeder.rec[count][1]==10||feeder.rec[count]
[1]==11||feeder.rec[count][1]==12||feeder.rec[count][1]==4||feeder.rec[count][1]==5)
        { device(feeder.rec[count][2],0);}
        count-- ;
        if(feeder.rec[count][1]==1||feeder.rec[count][1]==2||
feeder.rec[count][1]==3)
          { if(feeder.rec[count][3]==0) stop=1; }
    }

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        check_load();
    }while(stop!=1);
}
feeder.rec[16][4] = 3;    /* default feeder */
for(i=0;i<=15;i++)
    { feeder.rec[i][4] = 1; }
for(i=17;i<=32;i++)
    { feeder.rec[i][4] = 2; }
}

/*****/
device (int sno,int status)
/*****/
{
int x,y,active;
x = sno/10;
y = fmod(sno,10); if (y == 0) y = 10;
if (y == 10) x = x-1 ;
x = xdim*x ; y = ydim*y;
active = feeder.rec[ screen.rec[sno][1] ][1];
if( feeder.rec[screen.rec[sno][1]][1]==1||feeder.rec[screen.rec[sno][1]]
[1]==2||feeder.rec[screen.rec[sno][1]][1]==3)
    { sw_sound();
      if (status==1||status == 2) status = 3;}
if (status == 0)
    { display_symbol (x,y,active,0);
      feeder.rec[screen.rec[sno][1]][3] = 0;}
if (status == 1)
    { display_symbol (x,y,active,3);
      feeder.rec[screen.rec[sno][1]][3] = 1;}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    if (status == 2)
        { display_symbol (x,y,active,2);
          feeder.rec[screen.rec[sno][1]][3] = 1;}
    if (status == 3)
        { display_symbol (x,y,active,1);
          feeder.rec[screen.rec[sno][1]][3] = 1;}
}

```

```

/*****/
display_symbol(int startx,int starty,int npic,int status)
/*****/
{
    int x,y ;
    if (status==1)
        for(y = starty; y < starty+ydim; y++)
            for(x = startx; x < startx+xdim; x++) /* normol red for sw.*/
                putpixel(x,y,sym.image[npic][x-startx][y-starty]);
    if (status==2)
        for(y = starty; y < starty+ydim; y++)
            for(x = startx; x < startx+xdim; x++) /* green for f2*/
                if(sym.image[npic][x-startx][y-starty] == 4)
                    putpixel(x,y,2);
            else
                putpixel(x,y,sym.image[npic][x-startx][y-starty]);
    if (status==3)
        /* blue for f1*/
        for(y = starty; y < starty+ydim; y++)
            for(x = startx; x < startx+xdim; x++)
                if(sym.image[npic][x-startx][y-starty] == 4)
                    putpixel(x,y,1);
            else

```

```

        putpixel(x,y,sym.image[npic][x-startx][y-starty]);
if (status==0)
    for(y = starty; y < starty+ydim; y++)
        for(x = startx; x < startx+xdim; x++)
            if(sym.image[npic][x-startx][y-starty] == 4)
                putpixel(x,y,0);
            else
                putpixel(x,y,sym.image[npic][x-startx][y-starty]);
if (status==15)
    for(y = starty; y < starty+ydim; y++)
        for(x = startx; x < startx+xdim; x++)
            if(sym.image[npic][x-startx][y-starty] == 4)
                putpixel(x,y,15);
            else
                putpixel(x,y,sym.image[npic][x-startx][y-starty]);
if (status==14)
    for(y = starty; y < starty+ydim; y++)
        for(x = startx; x < startx+xdim; x++)
            if(sym.image[npic][x-startx][y-starty] == 4)
                putpixel(x,y,14);
            else
                putpixel(x,y,sym.image[npic][x-startx][y-starty]);
}

/*****/
back_ground()
/*****/

{
    setfillstyle(1,7);
    bar(0,0,639,479);

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

setfillstyle(1,11);
bar(0,0,639,30);
setcolor(1);
display_symbol(11,0,17,1);
display_symbol(83,0,17,1);
display_symbol(220,0,17,1);
display_symbol(343,0,17,1);
display_symbol(484,0,18,1);
display_symbol(564,0,17,1);
outtextxy(8,10,"    NEW    CLEAR FAULT    INPUT DATA    OPERRATE SW
MOVE    QUIT ");
setcolor(4);
outtextxy(8,10," F1    F2    F3    F10    F7    ");
outtextxy(89,43,"CB1");
outtextxy(488,403,"CB2");
setfillstyle(1,1);
/* bar(0,ydim*14-20,xdim*16,ydim*16);
display_symbol(3,ydim*14-10,1,0);
display_symbol(3,ydim*15-5,1,1);
display_symbol(xdim*6,ydim*14-10,2,0);
display_symbol(xdim*6,ydim*15-5,2,1);
display_symbol(xdim*11,ydim*14-10,9,0);
display_symbol(xdim*11,ydim*15-5,9,1);
setcolor(14);
outtextxy(xdim+5,ydim*14,":Opened circuit breaker");
outtextxy(xdim+5,ydim*15+4,":Closeed circuit breaker");
outtextxy(xdim*7+5,ydim*14,":Opened switch");
outtextxy(xdim*7+5,ydim*15+4,":Closeed switch");
outtextxy(xdim*12+5,ydim*14,":De-energized line");
outtextxy(xdim*12+5,ydim*15+4,":Energizeed line");
outtextxy(5,ydim*4-15,"CB-1");

```

```

outtextxy(5,ydim*10-15,"CB-2");
display_symbol(xdim,ydim*12,16,1); /*pic 16*/
/*
setcolor(1);
circle(xdim+19,ydim*12+15,19);
setttextstyle(1,0,2);
setfillstyle(1,14);
bar(xdim*2,ydim*12,xdim*15,ydim*13 );
setcolor(5);
outtextxy(xdim*2+10,ydim*12+5,"SIMULATION OF DISTRIBUTION AUTOMATIC
SYSTEM" );*/
}

/*****/
drow_picture ()
/*****/
{
display_symbol (xdim*2,ydim*2,1,1);
display_symbol (xdim*3,ydim*2,9,3);
display_symbol (xdim*4,ydim*2,2,1);
display_symbol (xdim*5,ydim*2,9,3);
display_symbol (xdim*6,ydim*2,2,1);
display_symbol (xdim*7,ydim*2,11,3);
/*Line3*/
display_symbol (xdim*12,ydim*14,0,1);
display_symbol (xdim*8,ydim*14,2,1);
display_symbol (xdim*9,ydim*14,9,3);
display_symbol (xdim*10,ydim*14,2,1);
display_symbol (xdim*11,ydim*14,9,3);
/*line 2 */
display_symbol (xdim*7,ydim*3,10,3);

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

display_symbol (xdim*7,ydim*4,3,1);
display_symbol (xdim*7,ydim*5,10,3);
display_symbol (xdim*7,ydim*6,3,1);
display_symbol (xdim*7,ydim*7,10,3);
display_symbol (xdim*7,ydim*8,3,15);
display_symbol (xdim*7,ydim*9,10,3);
display_symbol (xdim*7,ydim*10,3,1);
display_symbol (xdim*7,ydim*11,10,3);
display_symbol (xdim*7,ydim*12,3,1);
display_symbol (xdim*7,ydim*13,10,3);
display_symbol (xdim*7,ydim*14,12,3);
}
/*****/
init_data ()
/*****/
{ int i;
  for(i=0;i<=170;i++)
    { screen.rec[i][1]=100;}
/*1 = pic 2 = screen_no 3=status (0=off,1=on ) 4 =feeder 5 = fault_position */
feeder.rec[0][1]=1; feeder.rec[0][2]=13; feeder.rec[0][3]=1; feeder.rec[0][4]=1;
feeder.rec[1][1]=9; feeder.rec[1][2]=23; feeder.rec[1][3]=1; feeder.rec[1][4]=1;
feeder.rec[2][1]=2; feeder.rec[2][2]=33; feeder.rec[2][3]=1; feeder.rec[2][4]=1;
feeder.rec[3][1]=9; feeder.rec[3][2]=43; feeder.rec[3][3]=1; feeder.rec[3][4]=1;
feeder.rec[4][1]=2; feeder.rec[4][2]=53; feeder.rec[4][3]=1; feeder.rec[4][4]=1;
feeder.rec[5][1]=9; feeder.rec[5][2]=63; feeder.rec[5][3]=1; feeder.rec[5][4]=1;
feeder.rec[6][1]=2; feeder.rec[6][2]=73; feeder.rec[6][3]=1; feeder.rec[6][4]=1;
feeder.rec[7][1]=9; feeder.rec[7][2]=83; feeder.rec[7][3]=1; feeder.rec[7][4]=1;
feeder.rec[8][1]=2; feeder.rec[8][2]=93; feeder.rec[8][3]=1; feeder.rec[8][4]=1;
feeder.rec[9][1]=9; feeder.rec[9][2]=103; feeder.rec[9][3]=1; feeder.rec[9][4]=1;
feeder.rec[10][1]=2; feeder.rec[10][2]=113; feeder.rec[10][3]=1; feeder.rec[10][4]=1;
feeder.rec[11][1]=9; feeder.rec[11][2]=123; feeder.rec[11][3]=1; feeder.rec[11][4]=1;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

feeder.rec[12][1]=2; feeder.rec[12][2]=133; feeder.rec[12][3]=1; feeder.rec[12][4]=1;
feeder.rec[13][1]=11; feeder.rec[13][2]=143; feeder.rec[13][3]=1; feeder.rec[13][4]=1;
feeder.rec[14][1]=10; feeder.rec[14][2]=144; feeder.rec[14][3]=1; feeder.rec[14][4]=1;
feeder.rec[15][1]=5; feeder.rec[15][2]=145; feeder.rec[15][3]=1; feeder.rec[15][4]=1;
feeder.rec[16][1]=3; feeder.rec[16][2]=146; feeder.rec[16][3]=0; feeder.rec[16][4]=3;
feeder.rec[17][1]=5; feeder.rec[17][2]=147; feeder.rec[17][3]=1; feeder.rec[17][4]=2;
feeder.rec[18][1]=10; feeder.rec[18][2]=148; feeder.rec[18][3]=1; feeder.rec[18][4]=2;
feeder.rec[19][1]=12; feeder.rec[19][2]=149; feeder.rec[19][3]=1; feeder.rec[19][4]=2;
feeder.rec[20][1]=2; feeder.rec[20][2]=139; feeder.rec[20][3]=1; feeder.rec[20][4]=2;
feeder.rec[21][1]=4; feeder.rec[21][2]=129; feeder.rec[21][3]=1; feeder.rec[21][4]=2;
feeder.rec[22][1]=2; feeder.rec[22][2]=119; feeder.rec[22][3]=1; feeder.rec[22][4]=2;
feeder.rec[23][1]=4; feeder.rec[23][2]=109; feeder.rec[23][3]=1; feeder.rec[23][4]=2;
feeder.rec[24][1]=2; feeder.rec[24][2]=99; feeder.rec[24][3]=1; feeder.rec[24][4]=2;
feeder.rec[25][1]=4; feeder.rec[25][2]=89; feeder.rec[25][3]=1; feeder.rec[25][4]=2;
feeder.rec[26][1]=2; feeder.rec[26][2]=79; feeder.rec[26][3]=1; feeder.rec[26][4]=2;
feeder.rec[27][1]=4; feeder.rec[27][2]=69; feeder.rec[27][3]=1; feeder.rec[27][4]=2;
feeder.rec[28][1]=2; feeder.rec[28][2]=59; feeder.rec[28][3]=1; feeder.rec[28][4]=2;
feeder.rec[29][1]=4; feeder.rec[29][2]=49; feeder.rec[29][3]=1; feeder.rec[29][4]=2;
feeder.rec[30][1]=2; feeder.rec[30][2]=39; feeder.rec[30][3]=1; feeder.rec[30][4]=2;
feeder.rec[31][1]=4; feeder.rec[31][2]=29; feeder.rec[31][3]=1; feeder.rec[31][4]=2;
feeder.rec[32][1]=1; feeder.rec[32][2]=19; feeder.rec[32][3]=1; feeder.rec[32][4]=2;

```

```

/* position on screen */

```

```

screen.rec[13][1]=0;
screen.rec[23][1]=1;
screen.rec[33][1]=2;
screen.rec[43][1]=3;
screen.rec[53][1]=4;
screen.rec[63][1]=5;
screen.rec[73][1]=6;
screen.rec[83][1]=7;
screen.rec[93][1]=8;

```

```

screen.rec[103][1]=9;
screen.rec[113][1]=10;
screen.rec[123][1]=11;
screen.rec[133][1]=12;
screen.rec[143][1]=13;
screen.rec[144][1]=14;
screen.rec[145][1]=15;
screen.rec[146][1]=16;
screen.rec[147][1]=17;
screen.rec[148][1]=18;
screen.rec[149][1]=19;
screen.rec[139][1]=20;
screen.rec[129][1]=21;
screen.rec[119][1]=22;
screen.rec[109][1]=23;
screen.rec[99][1]=24;
screen.rec[89][1]=25;
screen.rec[79][1]=26;
screen.rec[69][1]=27;
screen.rec[59][1]=28;
screen.rec[49][1]=29;
screen.rec[39][1]=30;
screen.rec[29][1]=31;
screen.rec[19][1]=32;
}

```

```

/*****/

```

```

    sw_sound()

```

```

/*****/

```

```

{

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    sound(1318); delay(10); sound(1568);delay(10);
    sound(1975); delay(10);
    nosound();
}
/*****/
    fault_sound()
/*****/
{ int f;
    for(f = 0; f<1000 ;f ++)
        { sound(20*f);
          delay(1);}
    nosound();
}
/*****/
    getkey()
/*****/
{
    union key {
    int i;
    char ch[2];
    }k;
    k.i=bioskey(0);
    return k.ch[1];
}

/*****/
load_symbol(void)
/*****/

{ FILE *fp;
    int result;

```

```

fp = fopen("das.pic", "rb");
fread(&sym, sizeof sym, 1, fp);
if(ferror(fp)) result = 0;
else result = 1;
fclose(fp);
return result;
}

```

```

/*****/
int mouse_press(int *mro, int *mco, int button)
/*****/
{
    union REGS ireg;
    ireg.x.ax = 0x05;          /* Function to get mouse data */
    ireg.x.bx = button;      /* Update only on this button press */
    int86(0x33, &ireg, &ireg);
    *mco = ireg.x.cx;
    *mro = ireg.x.dx;
    return (ireg.x.ax);      /* Return button status */
}
/*****/
int mouse_release(int *mro, int *mco, int button)
/*****/
{
    union REGS ireg;
    ireg.x.ax = 0x06;          /* Function to get mouse data */
    ireg.x.bx = button;      /* The button to inspect */
    int86(0x33, &ireg, &ireg);
    *mco = ireg.x.cx;
    *mro = ireg.x.dx;
    return (ireg.x.ax);      /* Return button status */
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

}
/*****/
mouse_counters(int *mro, int *mco)
/*****/
{
    union REGS ireg;
    ireg.x.ax = 0x0b;          /* Function to read motion counts */
    int86(0x33, &ireg, &ireg);
    *mco = ireg.x.cx;
    *mro = ireg.x.dx;
}

/*****/
int mouse_status(int *buttons)
/*****/
{
    union REGS ireg;
    ireg.x.ax = 0x00;
    int86(0x33, &ireg, &ireg);
    *buttons = ireg.x.bx;
    return ireg.x.ax;
}

/*****/
input_data()
/*****/
{
    float sys_data [4];
    int posi1 [4][2]={ {28,10},{41,12},{30,14},{30,16} };
    int posi2 [21][2]={ {24,5},{43,5},{59,5},

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        {24,6},{43,6},{59,6},
        {24,7},{43,7},{59,7},
        {24,8},{43,8},{59,8},
        {24,9},{43,9},{59,9},
        {24,10},{43,10},{59,10},
        {24,11},{43,11},{59,11},
    };

int posi3 [21][2]={ {24,17},{43,17},{59,17},
                    {24,18},{43,18},{59,18},
                    {24,19},{43,19},{59,19},
                    {24,20},{43,20},{59,20},
                    {24,21},{43,21},{59,21},
                    {24,22},{43,22},{59,22},
                    {24,23},{43,23},{59,23},
                    };

int x,y,i,j,k,l,set,fault_point;
char ch,c,temp[20];
    for(i=0;i<=13;i++)
        { for(j=1;j<=3;j++)
            f_data[i][j]=0;}
    for(i=0;i<=3;i++)
        sys_data[i]=0;

clrscr();
textmode(3);
textcolor(14);
box(8,5,68,17);
textbackground (4);
textcolor(14);
gotoxy(35,7);cprintf("INPUT DATA");
textmode(-1);
textcolor(2);

```

```

gotoxy(10,10);cprintf("SYSTEM VOLTAGE :      (KV)");
gotoxy(10,12);cprintf("CAPACITY OF EGAT TRANSFORMER :      (MVA)");
gotoxy(10,14);cprintf("FAULT AT BUS I-3 :      (A)");
gotoxy(10,16);cprintf("FAULT AT BUS I-g :      (A)");
textmode(3);
textbackground (4);
textcolor(1);
gotoxy(20,20);cprintf("[Enter]");
gotoxy(37,20);cprintf("[Esc]");
textmode(-1);
textcolor(2);
gotoxy(28,20);cprintf("to move");
gotoxy(43,20);cprintf("to quit");
j = 0; i=0;
do {
    gotoxy(posi1[j][0]+i,posi1[j][1]);
    ch=getch();
    if((ch >='0'&& ch <='9')||ch=='.')
        { putchar(ch);
          temp[i]=ch;
          i++;
        }
    else
        if(ch=='\b'||ch==75)
            { gotoxy(posi1[j][0]+(i-1),posi1[j][1]);
              printf(" ");
              i--;if(i < 0)i=0;
            }
        else
            if(ch==13||ch==72||ch==80)
                { if(i >0 )

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดก็ตาม ห้ามนำไปดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        { temp[i]='\0';
          sys_data[j] = atof(temp);}
        gotoxy(posi1[j][0],posi1[j][1]);
        printf("      ");
        gotoxy(posi1[j][0],posi1[j][1]);
        printf("%4.2f",sys_data[j]);
        if(ch==13||ch==80)j++;
        if(j > 3){j=0;}
        if(ch==72)j--;
        if(j < 0) j=0;i=0;
    }
    }while(ch != 27);
    system_v = sys_data[0];
    trans_cap = sys_data[1];
    fault_3 = sys_data[2];
    fault_g = sys_data[3];
clrscr();
textcolor(14);
box(8,1,78,10);
/*textcolor(4);*/
textbackground (1);
textcolor(15);
gotoxy(10,2); cprintf("[ FEEDER NO 1 ]");
textmode(-1);
textcolor(2);
gotoxy(40,2); cprintf("(Esc : to quit F1 : conductor code)");
textcolor(3);
gotoxy(10,3); cprintf("CB 1 CAPACITY (KVA):");
textcolor(14);
gotoxy(10,4); cprintf(" NODE NO.  INTERVAL(Km)  CONDUCTOR_CODE
INSTALLED(KVA)");

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

textcolor(2);
gotoxy(10,5); cprintf(" 1");
gotoxy(10,6); cprintf(" 2");
gotoxy(10,7); cprintf(" 3");
gotoxy(10,8); cprintf(" 4");
gotoxy(10,9); cprintf(" 5");
/* gotoxy(10,10);cprintf(" 6");
   gotoxy(10,11);cprintf(" 7");*/
gotoxy(31,3);scanf("%f",&cb1_cap);
j = 0; i=0; k=0; l=0; set=1;
do {
    gotoxy(posi2[j][0]+i,posi2[j][1]);
    ch=getch();
    if((ch >='0' && ch <='9')||ch=='.')
        { putchar(ch);
          temp[i]=ch;
          i++;}
    else
        if(ch=='\b'||ch==75)
            { gotoxy(posi2[j][0]+(i-1),posi2[j][1]);
              printf(" ");
              i--;if(i < 0 ) i = 0; }
        else
            if(ch==13||ch==80)
                { if(set==0){conductor_code(0);set =1;}
                  if(i >0 )
                      { temp[i]='\0';
                        f_data[k][l] = atoi(temp);
                      }
                  gotoxy(posi2[j][0],posi2[j][1]);
                  printf(" ");

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

gotoxy(posi2[j][0],posi2[j][1]);
printf("%4d",f_data[k][l]);
    j++; l++;
if(l > 2) { k++; l=0; }
if(k > 6) { k=0;}
if(j >14) { j=0; }
i = 0;
}
else
if(ch==59)
    if(set==1)
        {conductor_code(1);set = 0;}
    else
        {conductor_code(0);set = 1;}
} while ( ch != 27);
textcolor(14);
box(8,13,78,22);
/*textcolor(2);*/
textbackground (2);
textcolor(15);
gotoxy(10,14); cprintf("[ FEEDER NO 2 ]");
textmode(-1);
textcolor(2);
gotoxy(40,14); cprintf("(Esc : to quit F1 : conductor code)");
textcolor(3);
gotoxy(10,15); cprintf("CB 2 CAPACITY (KVA):");
textcolor(14);
gotoxy(10,16); cprintf(" NODE NO. INTERVAL(Km) CONDUCTOR_CODE
INSTALLED(KVA)*);
textcolor(2);
gotoxy(10,17); cprintf(" 1");

```

```

gotoxy(10,18); cprintf(" 2");
gotoxy(10,19); cprintf(" 3");
gotoxy(10,20); cprintf(" 4");
gotoxy(10,21); cprintf(" 5");
/* gotoxy(10,22); cprintf(" 6");
   gotoxy(10,23); cprintf(" 7");*/
gotoxy(31,15);scanf("%f",&cb2_cap);
j = 0; i=0; k=13; l=0; set=1;
do {
    gotoxy(posi3[j][0]+i,posi3[j][1]);
    ch=getch();
    if((ch >='0' && ch <='9')||ch=='.')
        { putchar(ch);
          temp[i]=ch;
          i++;}
    else
        if(ch=='\b'||ch==75)
            { gotoxy(posi3[j][0]+(i-1),posi3[j][1]);
              printf(" ");
              i--;if(i < 0 ) i = 0; }
        else
            if(ch==13||ch==80)
                { if(set==0){conductor_code(0);set =1;}
                  if(i >0 )
                      { temp[i]='\0';
                        f_data[k][l] = atoi(temp);
                      }
                  gotoxy(posi3[j][0],posi3[j][1]);
                  printf(" ");
                  gotoxy(posi3[j][0],posi3[j][1]);

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        printf("%4d",f_data[k][l]);
        j++;l++;
        if(l > 2) { k--; l=0; }
        if(k < 7) { k=13;}
        if(j >14) { j=0; }
        i = 0;
    }
else
    if(ch==59)
        if(set==1)
            {conductor_code(1);set = 0;}
        else
            {conductor_code(0);set = 1;}
    } while ( ch != 27);
}
/*****/
calculate_fault()
/*****/
{
    float i_base,z_base,z,z1,tz1,z120,tz120,con,dis,r,x1,r0,x0,angle,i_3,i_g,
          z3f,i_gmin,tz120f;
    float temp_r , temp_x ,temp_r0,temp_x0;
    int i,j,k,l,fault_point,fd,start;
    char tbuff [8];

    /* R ohm/Km X1 = X2 R0 + J0 */
    float table [16][5]={ {1, 1.274624,0.401496,1.422624,1.628310},
                          {2, 0.919246,0.391226,1.067246,1.618040},
                          {3, 0.640146,0.379854,0.788146,1.606667},
                          {4, 0.472079,0.366401,0.620079,1.593214},
                          {5, 0.340589,0.356424,0.488589,1.583237},
                          {6, 0.266435,0.348692,0.414432,1.575505},

```

```

    {7, 0.175713,0.334439,0.323713,1.561251},
    {8, 1.071094,0.391226,1.219094,1.618040},
    {9, 0.743943,0.379854,0.891943,1.606667},
    {10,0.562266,0.366401,0.710266,1.593214},
    {11,0.396144,0.356424,0.544144,1.583237},
    {12,1.348244,0.398347,1.496244,1.625161},
    {13,0.936331,0.386891,1.084331,1.613705},
    {14,0.666683,0.376216,0.814683,1.603030},
    {15,0.463186,0.356954,0.611186,1.583767},
    {16,0.343081,0.347499,0.491081,1.574312}
};

run = screen.rec[recono][1]; /* get fault point */
if(run==13||run==14||run==15){ run = 13;}
if(run==17||run==18||run==19){ run = 19;}
if(feeder.rec [run][4]==1)
    { fd = 1; start = 0; fault_point = (run-1)/2;}
if(feeder.rec [run][4]==2)
    { fd = 2; start = 13;fault_point = (run-5)/2;}
    /* calculate fault */
    i_base = trans_cap*1000/( system_v*sqrt(3) );
    z_base = pow(system_v,2)/trans_cap;
    z1 = (1*i_base)/fault_3;
    z120=(3*1*i_base)/fault_g;

    /****** cal three-phase fault *****/
temp_r =0;temp_x =0;
if(fd == 1)
    { for(i = start;i <= fault_point;i++) /* cal ro,x1 any section */
        { con = f_data [i][1] ; dis = f_data [i][0];
            for(j = 0 ; j <= 16 ;j++) /* open table */
                { if(table[j][0] == con )
                    { r = table[j][1];

```

```

        x1 = table[j][2];
        j = 16; }
    }
    z = (sqrt((pow(r,2)+pow(x1,2)))*dis)/z_base;
    angle = atan(x1/r);
    temp_r = temp_r+(z * cos(angle));
    temp_x = temp_x+(z * sin(angle));
}
}
if(fd == 2)
{
    for(i = start;i >= fault_point;i--) /* cal ro,x1 any section */
    {
        con = f_data [i][1] ; dis = f_data [i][0];
        for(j = 0 ; j <= 16 ;j++) /* open table */
        {
            if(table[j][0] == con )
            {
                r = table[j][1];
                x1 = table[j][2];
                j = 16; }
            }
            z = (sqrt((pow(r,2)+pow(x1,2)))*dis)/z_base;
            angle = atan(x1/r);
            temp_r = temp_r+(z * cos(angle));
            temp_x = temp_x+(z * sin(angle));
        }
    }
}

r = temp_r;x1 = temp_x;
tz1 = sqrt(( pow(r,2)+pow((z1+x1),2)));
i_3 = (1*i_base)/tz1;

/****** cal fault phase-ground *****/
temp_r0 =0; temp_x0 =0;temp_r = 0;temp_x = 0;
if(fd == 1)

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

{ for(i = start;i <= fault_point;i++) /* cal r,x1,r0,x0 any section */
  { con = f_data [i][1] ; dis = f_data [i][0];
    for(j = 0 ; j <= 16 ;j++) /* open table */
      { if(table[j][0] == con )
        {r = table[j][1];
         x1 = table[j][2];
         r0 = table[j][3];
         x0 = table[j][4];
         j = 16; }
        }
      z = (sqrt((pow(r,2)+pow(x1,2))) * dis*2)/z_base;
      angle = atan(x1/r);
      temp_r = temp_r+(z * cos(angle));
      temp_x = temp_x+(z * sin(angle));
      z = (sqrt((pow(r0,2)+pow(x0,2))) * dis)/z_base;
      angle = atan(x0/r0);
      temp_r0 = temp_r0+(z * cos(angle));
      temp_x0 = temp_x0+(z * sin(angle));
    }
  }
if(fd == 2)
{ for(i = start;i >= fault_point;i--) /* cal r,x1,r0,x0 any section */
  { con = f_data [i][1] ; dis = f_data [i][0];
    for(j = 0 ; j <= 16 ;j++) /* open table */
      { if(table[j][0] == con )
        {r = table[j][1];
         x1 = table[j][2];
         r0 = table[j][3];
         x0 = table[j][4];
         j = 16; }
        }

```

```

    }
    z = (sqrt((pow(r,2)+pow(x1,2)))*dis*2)/z_base;
    angle = atan(x1/r);
    temp_r = temp_r+(z * cos(angle));
    temp_x = temp_x+(z * sin(angle));
    z = (sqrt((pow(r0,2)+pow(x0,2)))*dis)/z_base;
    angle = atan(x0/r0);
    temp_r0 = temp_r0+(z * cos(angle));
    temp_x0 = temp_x0+(z * sin(angle));
}
}
/***** sum impedant *****/
r = temp_r+temp_r0; x1 = temp_x+temp_x0;
tz120 = sqrt(( pow(r,2)+pow((z120+x1),2)));
i_g = (3*i_base)/tz120;
z3f = 40/z_base;
tz120f = sqrt( pow((r+z3f),2)+pow((z120+x1) ,2) );
i_gmin = 3*i_base/tz120f;
/*gbox(xdim*7-8,ydim*5,xdim*13+8,ydim*7,12);*/
/*gbox(10,ydim*25,xdim*5+10,ydim*26,14);*/
settextstyle(0,0,1);
setcolor(0);
outtextxy(45,350,"FAULT CURRENT");
outtextxy(45,360,"THREE PHASE FAULT =");
outtextxy(45,370,"SINGLE LINE FAULT =");
outtextxy(45,380,"DOUBLE LINE FAULT =");
sprintf(tbuff,"%4.2f",i_3);
strcat(tbuff," A");
outtextxy(210,360,tbuff);
sprintf(tbuff,"%4.2f",i_g);
strcat(tbuff," A");

```

```

        outtextxy(210,370,tbuff);
        sprintf(tbuff,"%4.2f",i_gmin);
        strcat(tbuff," A");
        outtextxy(210,380,tbuff);
    } /* end calculate */
    /*****/
    gbox(int x,int y,int x1,int y1,int color)
    /*****/
    {
        setfillstyle(1,color);
        bar(x,y,x1,y1);
        setcolor(0);
        line(x,y,x1,y);line(x,y1,x1,y1);line(x,y,x,y1);line(x1,y,x1,y1);
    }
    /*****/
    box(int xul,int yul,int xlr,int ylr)
    /*****/
    {
        int i;
        char hz = 196;
        char vt = 179;
        char ul = 218;
        char ur = 191;
        char lr = 217;
        char ll = 192;

        gotoxy(xul,yul);
        for(i=xul;i<=xlr;i++)
            putch(hz);
        gotoxy(xul,ylr);
        for(i=xul;i<=xlr;i++)
            putch(hz);
    }

```

```

for(i=yul;i<=ylr;i++)
    { gotoxy(xul,i);
      putchar(vt);
      gotoxy(xlr,i);
      putchar(vt);}
gotoxy(xul,yul);putchar(ul);
gotoxy(xlr,yul);putchar(ur);
gotoxy(xlr,ylr);putchar(lr);
gotoxy(xul,ylr);putchar(ll);
}

/*****/
conductor_code(int status)
/*****/
{ char buf[19*20*2];
  int y;
  if(status == 1){
    gettext(63,3,80,21,buf);
    for(y=3;y<=21;y++)
      {gotoxy(64,y);printf("          ");}
    box(63,3,80,21);
    textmode(3);
    textbackground (15);
    textcolor(5);
    gotoxy(64,4); cprintf("CONDUCTOR : CODE");
    gotoxy(64,5); cprintf(" 25A  : 1 ");
    gotoxy(64,6); cprintf(" 35A  : 2 ");
    gotoxy(64,7); cprintf(" 50A  : 3 ");
  }
}

```

```

gotoxy(64,8); cprintf(" 70A : 4 ");
gotoxy(64,9); cprintf(" 95A : 5 ");
gotoxy(64,10);cprintf(" 120A : 6 ");
gotoxy(64,11);cprintf(" 185A : 7 ");
gotoxy(64,12);cprintf(" 35AAY : 8 ");
gotoxy(64,13);cprintf(" 50AAY : 9 ");
gotoxy(64,14);cprintf(" 70AAY : 10 ");
gotoxy(64,15);cprintf(" 95AAY : 11 ");
gotoxy(64,16);cprintf(" 25ACSR : 12 ");
gotoxy(64,17);cprintf(" 35ACSR : 13 ");
gotoxy(64,18);cprintf(" 50ACSR : 14 ");
gotoxy(64,19);cprintf(" 70ACSR : 15 ");
gotoxy(64,20);cprintf(" 95ACSR : 16 ");
textmode(-1);
}
if(status==0)
    puttext(63,3,80,21,buf);
}

/*****/
write_data()
/*****/

{ char tbuff [15];
  int i;

  settxtstyle(0,0,1);
  setcolor(0);/*1*/
  /*sprintf(tbuff,"%4.2f",system_v);
  setfillstyle(1,7);
  bar(xdim-20,ydim*6-8,xdim*5,ydim*6);
  outtextxy(xdim-20,ydim*6,"SYSTEM VOLTAGE");

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

strcat(tbuff, " KV");
outtextxy(xdim*4,ydim*6,tbuff); */
gbox(10,ydim*5,xdim*5+10,ydim*6,14);
gbox(10,ydim*6,xdim*5+10,ydim*7,11);
gbox(10,ydim*7,xdim*5+10,ydim*8,2);
gbox(xdim*1+10,ydim*5,xdim*3+10,ydim*6,14);
setcolor(0);
    outtextxy(15,ydim*6+10,"CB1");
    outtextxy(15,ydim*7+10,"CB2");
    outtextxy(xdim+10,ydim*5+10," CAPACITY");
    outtextxy(xdim+10,ydim*5+20," (KVA)");
gbox(xdim*3+10,ydim*5,xdim*5+10,ydim*6,14);
    outtextxy(xdim*3+10,ydim*5+10," LOAD");
    outtextxy(xdim*3+10,ydim*5+20," (KVA)");
/* gbox(xdim*5+10,ydim*5,xdim*7-15,ydim*6,14);
    outtextxy(xdim*5+10,ydim*5+10,"CURRENT");
    outtextxy(xdim*5+10,ydim*5+20," (A)");*/
sprintf(tbuff, "%4.2f",cb1_cap);
gbox(xdim*1+10,ydim*6,xdim*3+10,ydim*7,11);
outtextxy(xdim*1+20,ydim*6+10,tbuff);
sprintf(tbuff, "%4.2f",cb2_cap);
gbox(xdim*1+10,ydim*7,xdim*3+10,ydim*8,2);
outtextxy(xdim*1+20,ydim*7+10,tbuff);
setcolor(1);/*2*/
/* setfillstyle(1,7);
bar(xdim,ydim*2,xdim*13,ydim*2+20);
setfillstyle(1,7);
bar(xdim,ydim*10,xdim*13,ydim*10+20);*/
for(i=0 ;i<= 1 ;i++)
{ sprintf(tbuff, "%4d",f_data[i][2]);
    strcat(tbuff, " KVA");

```

```

    outtextxy(xdim*(3.7+(2*i)),ydim*1.3,tbuff);}
for(i=13 ;i>= 12 ;i--)
    { sprintf(tbuff,"%4d",f_data[i][2]);
      strcat(tbuff," KVA");
      outtextxy(xdim*((15.8-i)*2+2),ydim*15.2,tbuff);}
setcolor(6);/*6*/
for(i=0 ;i<= 1 ;i++)
    { sprintf(tbuff,"%4d",f_data[i][0]);
      strcat(tbuff," Km");
      outtextxy(xdim*(3.7+(2*i)),ydim*1.7,tbuff);}
for(i=13 ;i>= 12 ;i--)
    { sprintf(tbuff,"%4d",f_data[i][0]);
      strcat(tbuff," Km");
      outtextxy(xdim*((15.8-i)*2+2),ydim*15.5,tbuff);}
for(i=0 ;i<= 2 ;i++)
    { sprintf(tbuff,"%4d",f_data[i+2][0]);
      strcat(tbuff," Km");
      outtextxy(xdim*(8),ydim*(3.4+(2*i)),tbuff);}
for(i=3 ;i<= 5 ;i++)
    { sprintf(tbuff,"%4d",f_data[i+6][0]);
      strcat(tbuff," Km");
      outtextxy(xdim*(8),ydim*(3.4+(2*i)),tbuff);}
setcolor(1);
for(i=0 ;i<= 2 ;i++)
    { sprintf(tbuff,"%4d",f_data[i+2][2]);
      strcat(tbuff," KVA");
      outtextxy(xdim*(8),ydim*(3+(2*i)),tbuff);}
for(i=3 ;i<= 5 ;i++)
    { sprintf(tbuff,"%4d",f_data[i+6][2]);
      strcat(tbuff," KVA");
      outtextxy(xdim*(8),ydim*(3+(2*i)),tbuff);}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

}

/*****/
feeder_data()
/*****/

{
system_v = 22;trans_cap = 50;fault_3 = 4188;fault_g = 5624;
cb1_cap = 10000;cb2_cap = 10000;
/* distance    conductor code    insttall KVA */
f_data [0][0] = 5;f_data[0][1] = 7;f_data[0][2] = 1000;
f_data [1][0] = 5;f_data[1][1] = 7;f_data[1][2] = 1000;
f_data [2][0] = 5;f_data[2][1] = 7;f_data[2][2] = 1000;
f_data [3][0] = 5;f_data[3][1] = 7;f_data[3][2] = 1000;
f_data [4][0] = 5;f_data[4][1] = 7;f_data[4][2] = 1000;
f_data [5][0] = 5;f_data[5][1] = 7;f_data[5][2] = 1000;
f_data [6][0] = 5;f_data[6][1] = 7;f_data[6][2] = 1000;
f_data [7][0] = 5;f_data[7][1] = 7;f_data[7][2] = 1000;
f_data [8][0] = 5;f_data[8][1] = 7;f_data[8][2] = 1000;
f_data [9][0] = 5;f_data[9][1] = 7;f_data[9][2] = 1000;
f_data [10][0] = 5;f_data[10][1] = 7;f_data[10][2] = 1000;
f_data [11][0] = 5;f_data[11][1] = 7;f_data[11][2] = 1000;
f_data [12][0] = 5;f_data[12][1] = 7;f_data[12][2] = 1000;
f_data [13][0] = 5;f_data[13][1] = 7;f_data[13][2] = 1000;
}

/*****/
check_load()
/*****/

{

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

int count,chk;
int sw_chk[14]={1,3,5,7,9,11,14,18,21,23,25,27,29,31};
char tbuff [15];
lcb1=0 ; lcb2=0 ; count = 0; chk = sw_chk [count];
while(feeder.rec [chk][3] == 1 && feeder.rec[chk][4] ==1)
    { lcb1 = lcb1+f_data[count][2];
      count++;
      chk = sw_chk [count];
    }
count = 13; chk = sw_chk [count];
while(feeder.rec [chk][3] == 1 && feeder.rec[chk][4] ==2)
    { lcb2 = lcb2+f_data[count][2];
      count--;
      chk = sw_chk [count];
    }
settextstyle(0,0,1);
setcolor(0);/*1*/
printf(tbuff,"%5d",lcb1);
gbox(xdim*3+10,ydim*6,xdim*5+10,ydim*7,11);
outtextxy(xdim*4,ydim*6+10,tbuff);
printf(tbuff,"%5d",lcb2);
gbox(xdim*3+10,ydim*7,xdim*5+10,ydim*8,2);
outtextxy(xdim*4,ydim*7+10,tbuff);
}

/*****/
showfault()
/*****/

{
if (showf == 127)

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

{
  setfillstyle(SOLID_FILL,7);
  bar(44,297,280,390);
  outtextxy(50,300,"THREE PHASE FAULT");
  calculate_fault();
  display_symbol (xdim*3,ydim*2,9,3);
  display_symbol (xdim*4,ydim*2,2,2);
  display_symbol (xdim*5,ydim*2,9,2);
  display_symbol (xdim*6,ydim*2,2,2);
  display_symbol (xdim*7,ydim*2,11,3);
  /*Line3*/
  display_symbol (xdim*12,ydim*14,0,1);
  display_symbol (xdim*8,ydim*14,2,1);
  display_symbol (xdim*9,ydim*14,9,3);
  display_symbol (xdim*10,ydim*14,2,1);
  display_symbol (xdim*11,ydim*14,9,3);
  /*line 2 */
  display_symbol (xdim*7,ydim*3,10,3);
  display_symbol (xdim*7,ydim*4,3,1);
  display_symbol (xdim*7,ydim*5,10,3);
  display_symbol (xdim*7,ydim*6,3,1);
  display_symbol (xdim*7,ydim*7,10,3);
  display_symbol (xdim*7,ydim*8,3,14);
  display_symbol (xdim*7,ydim*9,10,3);
  display_symbol (xdim*7,ydim*10,3,1);
  display_symbol (xdim*7,ydim*11,10,3);
  display_symbol (xdim*7,ydim*12,3,1);
  display_symbol (xdim*7,ydim*13,10,3);
  display_symbol (xdim*7,ydim*14,12,3);
}
if (showf == 191)

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

{
setfillstyle(SOLID_FILL,7);
bar(44,297,280,390);
outtextxy(50,305,"SINGLE LINE TO GROUND FAULT");
calculate_fault();
display_symbol (xdim*3,ydim*2,9,3);
display_symbol (xdim*4,ydim*2,2,1);
display_symbol (xdim*5,ydim*2,9,3);
display_symbol (xdim*6,ydim*2,2,1);
display_symbol (xdim*7,ydim*2,11,3);
/*Line3*/
display_symbol (xdim*12,ydim*14,0,1);
display_symbol (xdim*8,ydim*14,2,1);
display_symbol (xdim*9,ydim*14,9,3);
display_symbol (xdim*10,ydim*14,2,1);
display_symbol (xdim*11,ydim*14,9,3);
/*line 2 */
display_symbol (xdim*7,ydim*3,10,3);
display_symbol (xdim*7,ydim*4,3,1);
display_symbol (xdim*7,ydim*5,10,3);
display_symbol (xdim*7,ydim*6,3,1);
display_symbol (xdim*7,ydim*7,10,3);
display_symbol (xdim*7,ydim*8,3,14);
display_symbol (xdim*7,ydim*9,10,3);
display_symbol (xdim*7,ydim*8,3,14);
display_symbol (xdim*7,ydim*10,3,2);
display_symbol (xdim*7,ydim*11,10,2);
display_symbol (xdim*7,ydim*12,3,2);
display_symbol (xdim*7,ydim*13,10,3);
display_symbol (xdim*7,ydim*14,12,3);
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

if (showf == 239) /*****F1-F2*****/
{
    setfillstyle(SOLID_FILL,7);
    bar(44,297,280,390);
    outtextxy(50,310," THREE PHASE FAULT");
    outtextxy(50,320," SINGLE LINE TO GROUND FAULT");
    calculate_fault();
    display_symbol (xdim*3,ydim*2,9,3);
    display_symbol (xdim*4,ydim*2,2,1);
    display_symbol (xdim*5,ydim*2,9,3);
    display_symbol (xdim*6,ydim*2,2,1);
    display_symbol (xdim*7,ydim*2,11,3);
    /*Line3*/
    display_symbol (xdim*12,ydim*14,0,1);
    display_symbol (xdim*8,ydim*14,2,2);
    display_symbol (xdim*9,ydim*14,9,2);
    display_symbol (xdim*10,ydim*14,2,2);
    display_symbol (xdim*11,ydim*14,9,3);
    display_symbol (xdim*7,ydim*8,3,14);
    /*line 2 */
    display_symbol (xdim*7,ydim*3,10,3);
    display_symbol (xdim*7,ydim*4,3,1);
    display_symbol (xdim*7,ydim*5,10,3);
    display_symbol (xdim*7,ydim*6,3,1);
    display_symbol (xdim*7,ydim*7,10,3);
    display_symbol (xdim*7,ydim*8,3,14);
    display_symbol (xdim*7,ydim*9,10,3);
    display_symbol (xdim*7,ydim*10,3,1);
    display_symbol (xdim*7,ydim*11,10,3);
    display_symbol (xdim*7,ydim*12,3,1);
    display_symbol (xdim*7,ydim*13,10,3);
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

display_symbol (xdim*7,ydim*14,12,3);
}
}
}

```



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การจัดการพลังงานในสถานีไฟฟ้าย่อย

Energy management in substation

อวิชชัย ธรรมประสิทธิ์* ผศ. มณฑล สิลลาจินดาทรกุล**

- * นักศึกษาปริญญาโท ภาควิชาวิศวกรรมไฟฟ้า คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
- ** ผู้ช่วยศาสตราจารย์ ภาควิชาวิศวกรรมไฟฟ้า คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

บทคัดย่อ. บทความวิจัยนี้ เป็นการเสนอวิธีการจัดการพลังงานในระดับสถานีไฟฟ้าย่อยและโครงข่ายของสถานีไฟฟ้าย่อยโดยใช้คอมพิวเตอร์เป็นตัวควบคุมปฏิบัติการปฏิบัติงานของสถานีไฟฟ้าย่อยในการถ่ายโอนโหลดในสายป้อนแต่ละสายหรือหม้อแปลงแต่ละตัวอาจเกิดโอเวอร์โหลดได้ ในขณะที่บางสายป้อนหรือหม้อแปลงบางตัวทำงานไม่ถึงพิกัด ดังนั้นบทความวิจัยนี้จึงมีแนวคิดที่จะทำการโอนถ่ายโหลดของสายป้อน หรือหม้อแปลงไปบางส่วนที่เกิดโอเวอร์โหลดไปยังส่วนที่สามารถรับโหลดเพิ่มได้โดยการเกิดแนวคิดการถ่ายโอนโหลดในลักษณะ HINN (HOPFIELD NEURAL NETWORK) บางส่วนมาช่วยในการประดิษฐ์ซอฟต์แวร์เพื่อใช้ควบคุมเซอร์กิตเบรกเกอร์ซึ่งเป็นเครื่องมือในการสวิตช์ของสถานีไฟฟ้าย่อยและโครงข่ายของสถานีไฟฟ้าย่อย

Abstract. This paper presents the application of energy management in substation and substation's network by controlling with PC. Generally, working in substation for transferring load in each feeder or transformer may overload, while some feeder or transformer don't work at the rating, therefore this paper has an idea to transfer some parts of load feeders or transformers that overload to sections which can increasingly obtain load. This can be made by using part of HINN (Hopfield Neural Network) idea combine with software in order to control circuit breaker that is an apparatus invention transferring in switching of substation and substation's network.

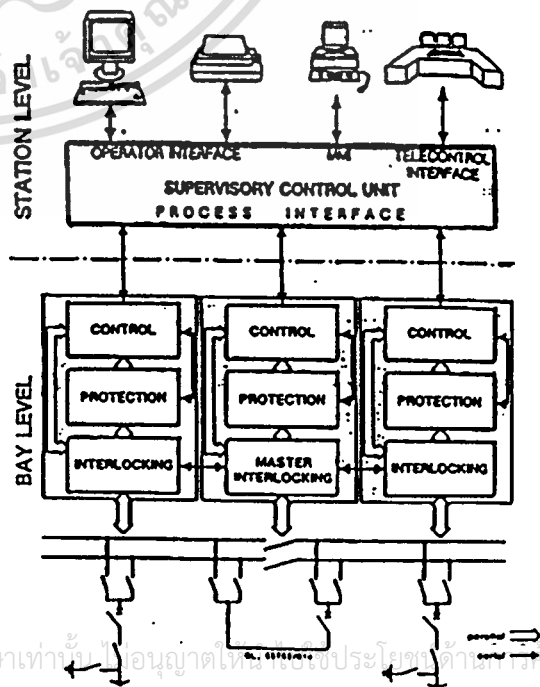
1. บทนำ

การใช้เครื่องคอมพิวเตอร์เข้ามาช่วยดูแล Substation ได้กำเนิดขึ้นในทศวรรษที่ 17 โดยในระยะเริ่มแรกจะใช้ Main frame computer ดูแลจะเป็นสิ่งพิมพ์และไม่ได้ทำหน้าที่การบำรุงรักษาเมื่อเปรียบเทียบกับเทคโนโลยีในปัจจุบัน การควบคุมที่พัฒนาอย่างรวดเร็วทางด้านคอมพิวเตอร์ในปัจจุบันจะควบคุม และป้องกันโดยการนำข้อมูลมาเชื่อมโยงกับที่ศูนย์กลาง (ซึ่งศูนย์กลางนี้จะเป็นตัวควบคุมและ ป้องกัน Substation ต่าง ๆ อีกทีหนึ่ง) การควบคุม และ ป้องกัน นี้จะมีประสิทธิภาพสูงสุดก็ต่อเมื่อมีอุปกรณ์ที่เพียงพอสำหรับการบริการข้อมูลจำนวนมากและอุปกรณ์ที่ทันสมัยระบบนี้จะประกอบไปด้วยส่วนควบคุมดูแล , ส่วนการ Interlock ของสวิตช์เกียร์ , Bay และ Busbar Protection รวมถึงอุปกรณ์ควบคุม Synchronized tap Changer, switching program ส่วนการรับแจ้งสัญญาณและ การควบคุมระยะไกล เป็นต้น

- 1) Supervisory Control
- 2) Inter lock and local Control
- 3) Protection System

จากรูปที่ 1.1 การควบคุมลักษณะนี้ จะต้องอาศัยการเชื่อมโยงข้อมูล Supervisory Control ในส่วนนี้จะประกอบไปด้วย อุปกรณ์ และ ระหว่างส่วนย่อยต่าง ๆ ของ Substation และระบบสายส่งโดยอาศัยศูนย์กลางซึ่งแต่ละส่วนสามารถควบคุมไม่ให้เกิดข้อผิดพลาดได้ Substation สามารถควบคุมได้โดย Operator อาจจะใช้ Manual หรือ Automatic จะมีการทบทวนเวลาขึ้นไปมากก็พอ และ สามารถแสดงผลได้ในจอภาพของคอมพิวเตอร์ที่อยู่ส่วนกลาง โดยแนวความคิดนี้พยายามที่จะทำให้มีอุปกรณ์น้อยที่สุดจึงได้ชื่อว่า Hierarchical

มาใช้ในการเชื่อมโยงข้อมูลทำให้มีการเชื่อมโยงข้อมูลได้มากขึ้นจนถึงมีการลด Noise ที่เกิดขึ้นได้อย่างมีประสิทธิภาพอีกด้วย



รูปที่ 1.1 ระบบควบคุมอัตโนมัติ

Supervisory Control ในส่วนนี้จะประกอบไปด้วย อุปกรณ์ และ เครื่องมือที่จำเป็นสำหรับการควบคุมระบบที่มีโครงข่ายใหญ่ โดยจะแบ่งเป็นส่วนต่าง ๆ ดังนี้

1.1 System control diagram เป็นส่วนที่ประกอบไปด้วย แผนภาพของระบบไฟฟ้าซึ่งมีส่วนของไฟฟ้าแรงสูง หม้อแปลงไฟฟ้า, Circuit Breaker, Converting Plant และ ทางเดินของ Cables ส่วนต่าง ๆ ของแผนภาพนี้จะมีสีแตกต่างกันเป็นการระบุถึงความแตกต่างของ Voltage

1.2 Telephone Equipment ในส่วนนี้จะเป็นการควบคุมอย่างมีประสิทธิภาพจากห้อง Control ไปยังส่วนที่ทำการ Operate จะต้องมีส่วนทำการเชื่อมโยงข้อมูลข่าวสาร นั่นก็คือส่วนของ Telephone Equipment นั้นเอง

1.3 Metering Equipment ในส่วนนี้เป็นการระบุและบันทึกค่าของ Load ในแต่ละส่วนของ Substation เป็นส่วนเริ่มแรกในการควบคุมในระดับ Substation โดยที่เครื่องมือในการวัด Voltage เป็นเครื่องมือที่ทำให้ส่วนควบคุมทราบถึงความเป็นไป ของระบบ เช่น Wattmeters และ Voltage Indication ก็เป็นส่วนหนึ่งของ Metering Equipment

1.4 Supervisory Control Equipment ในส่วนนี้จะเป็นส่วนที่ควบคุมระยะไกลของ Substation แบบ Automatic ซึ่งจะมีอุปกรณ์ที่ใช้ในการควบคุมส่วนกลางที่สำคัญ ดังนี้

- Circuit Breaker Indication (เป็นส่วนแสดงผลว่า Open or Close)
 - Feeder and Transformer Loads-Ampere,Kilowatts, Reactive KVA,Power Factor
 - Perfection Indication (เป็นส่วนแสดงผลการเชื่อมโยงของ Feeder ที่ใช้เปิด/ปิด จากส่วนกลาง)
 - Remote Control ของ Circuit Breakers
 - Remote Control ของ Transformer and Regulator Tapping
 - Remote Control ของ Transformer Parallel, Automatic and Non-Automatic Operation
 - Remote Control ของ Voltage Relay สำหรับ On Load Tap Change Transformers
 - ส่วนแสดงผลการทำงานของ Supervisory Control Unit
- นี่เป็นการเชื่อมโยงระหว่าง Telephone Wires และศูนย์ควบคุมส่ง การและสถานีไฟฟ้าที่อยู่ไกลออกไป อุปกรณ์เหล่านี้สามารถให้ Supply จาก 50 Volt Battery อุปกรณ์เหล่านี้จะประกอบไปด้วย ส่วนประกอบของ Relays และ Rotary Switches

2. การ Switching ใน Substation

การ Switching ใน Substation ที่มีอยู่ในปัจจุบันมักจะเป็นไปเพื่อป้องกันการเกิดเหตุผิดปกติ (Fault) ขึ้นใน Substation และในส่วนประกอบใน 1 หน่วยย่อยสำหรับการ Switching ดังรูปที่ 2.1

ในปัจจุบันการ Switching จะเป็นไปเพื่อการป้องกัน เมื่อมีความผิดปกติเกิดขึ้นซึ่งการ Switching นี้จะมีผลทำให้ระบบ Power System ของ Substation มี Reliability มากขึ้น แต่เราได้นำแนวคิดแบบ Hopfield Neural Network เพื่อจะทำให้ระบบ Power System ที่เป็น Network ที่มีประสิทธิภาพและมี Reliability ที่สูงขึ้น การเกิด fault ที่เกิดขึ้นใน Substation มีขนาด 66 KV ขึ้นไป การที่จะแก้ปัญหาการเกิด fault จะต้องมีการ Feeder ที่เชื่อมโยงถึงกันโดยมี Circuit Breaker เป็นตัว ON-OFF จึงจะทำให้เกิด Reliability ขึ้นในระบบและระบบจะมี Reliability เกิดขึ้นได้จะต้องมี Engineering Knowledge และประสบการณ์ที่ดี ดังนั้นการคำนวณในปัจจุบัน จึงมีการใช้ PC (Personal Computer) มาช่วยกับวิธี Hopfield Model (เพื่อการควบคุมที่มีประสิทธิภาพ ดังนั้นเมื่อเกิด Fault ขึ้นในระบบ Transmission line เราควรจะรักษา reliability ของระบบโดยอาศัย Convexity

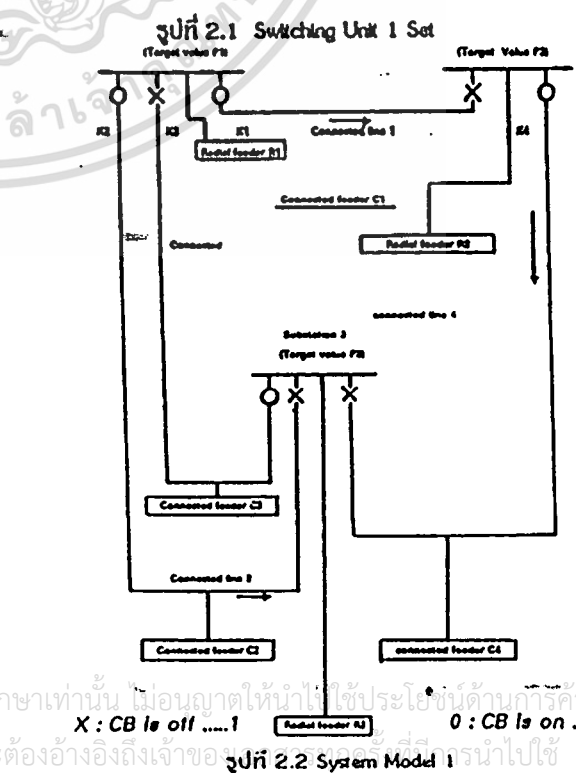
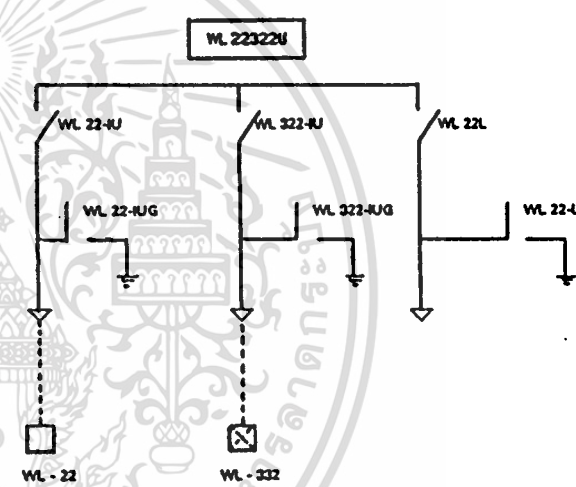
feeder จาก Substation ขึ้นนำมาทำการ switching โดยใช้ Circuit breaker ซึ่งการตัดสินใจ Operate จะเป็นปัญหาที่เป็น Optimization Problem ซึ่งมีวัตถุประสงค์ 2 ประการดังนี้

2.1 ค่า Error ที่เกิดขึ้นระหว่างค่าจุดประสงค์ที่เราตั้งไว้ (Target Value) และค่าของ Actual load ที่เกิดขึ้นจริงนั้นควรจะมีความต่ำที่สุด

2.2 สภาพของ Load ที่แท้จริงที่เกิดขึ้นนั้น จะต้องไม่เกินค่า Target Value ที่ตั้งไว้เกินกว่าขอบเขตที่เป็นไปได้โดยค่า Target Value คือการทาค่าของระบบให้มี reliability ที่ดีโดยอาจได้จาก Output ที่มีค่า 120 % ของผลรวมซึ่งค่า output จาก Power Supply (Transformer) ในการคำนวณจะเป็นดังนี้

ตัวอย่างเช่น ที่ Substation ระบบ 3 Phase Transformer 300 MVA (2 Bank) + 200 (1 Bank) เมื่อ Power factor มีค่าเท่ากับ 0.95 ดังนั้นจะได้

Target Value = (300 x (2-1)+200 x 1) x 0.95 x 1.2 และจากรูปที่ 2.2



จากรูปที่ 2.2 Transmission Line จะประกอบด้วย Circuit Breaker ที่มีสถานะ 0 และ 1 โดยจำนวน Connecting เป็น 2 ถัดมา 4 โดยมี Radial feeder R ได้รับพลังงานจาก Substation I และให้ค่า G เป็น ค่า Error ของสมการ $F = Cost Function$ เราจะได้ค่า Minimization Problem ดังสมการ

$$\begin{matrix} G1 & P1 & R1 & C1 \cdot C2 \cdot C3 & C1 & C2 & C3 & 0 & X1 \\ G2 & P2 & R2 & C4 & -C1 & 0 & 0 & C4 & X2 \dots(1) \\ G3 & P3 & R3 & 0 & 0 & -C2 & -C3 & -C4 & X3 \\ & & & & & & & & X4 \end{matrix}$$

จากสมการที่ (1) จะได้

$$G = P1 - R1 - S1 + \sum_{i=1}^4 A_i X_i \quad (i=1-3) \quad \dots\dots\dots(2)$$

$$F = \sum_{i=1}^3 G_i \quad \dots\dots\dots(3)$$

เมื่อ

- I = จำนวน Substation (I = 1-3)
- J = จำนวน Connected Feeder (J = 1-4)
- Xj = ค่าการเปลี่ยนแปลงของการ Switching Feeder J (0 หรือ 1), (0 <= Xj <= 1)
- P1 = Target Value ของ Substation I (MW)
- R1 = Radial Feeder Load ของ Substation I (MW)
- S1 = Power of Connected feeder ที่ Substation (MW)
- Cj = Load of Connected Feeder J (MW)
- Aj = Power of Connected Feeder J : โดยได้รับจาก Substation I

ค่าของ Cj จะเป็นลบสำหรับ Substation ที่มีขนาดเล็ก และมีค่าเป็นบวกเมื่อ Substation ที่มีขนาดเล็กจึงจะสอดคล้องกับสมการที่(1) ส่วนผลที่เป็นบวก ในส่วนแถว Aj จะถูกกำหนดโดย S1 และเราสามารถหาค่าสัมบูรณ์ของสมการที่ (3) จะเป็น Solution ของ HNN โดย F' ในสมการที่ (4) จะเป็นการตรวจสอบค่า Cost Function ว่าผลที่ดีที่สุดของ HNN ที่ได้นั้นมีค่าเป็นอย่างไร

$$F' = \sum_{i=1}^3 | P_i - R_i - S_i + \sum_{j=1}^4 A_j Y_j | \quad \dots\dots\dots(4)$$

โดย Yj = Round of value ของ Xj (ซึ่งมีค่า 0 หรือ 1)

หลักการสำหรับการ switching

การ Switching ใน Substation มีหลักในการ Switching ใน 1 หน่วยดังนี้

- พยายามทำให้ค่า Error of Substation (Err) มีค่ามากกว่าศูนย์ หรือ Target Value of Substation (Tgy) มากกว่า Actual Load (Act) ป้องกันไม่ไห้สถานะไฟฟ้าย่อยจ่ายโหลดเกิน พิกัดเป้าหมายของมันเอง ทำให้ได้โดยการตรวจว่าในสถานีไฟฟ้าย่อยนั้น มีค่าผลต่างของค่าเป้าหมายกับค่าแท้จริงมีค่าเป็นบวก ถ้าค่า Err of Substation ที่สถานีไฟฟ้าย่อยนั้นมีค่าน้อยกว่าศูนย์ เราจะต้อง ทำการเลือกหาสถานีไฟฟ้าย่อยอื่นที่มีค่าผลต่างของ Tgt กับ Act ที่เป็นบวกมาช่วยรับ connected Load ที่ต้องร่วมกับอยู่

ดังตัวอย่างที่กล่าวมาแล้ว สำหรับการใช้งานเพื่อการเลือกสถานีไฟฟ้าที่จะช่วยรับโหลดเกินอยู่

แสดงว่า สถานีไฟฟ้าบนจ่ายโหลดเกินอยู่มากเท่านั้น ซึ่งมีการจ่ายโหลดเกินมาก ๆ ส่วนจากการใช้งานซึ่งสืบลง อีกทั้งค่าประสิทธิภาพที่จะน้อยลงตามอัตราค่าที่คิดซึ่งมีการจ่ายโหลดเกินผิดพลาด ๆ ซึ่งทำให้เสียค่าใช้จ่ายมากตามไปด้วย

- ทดงจะมีการตรวจสอบสถานะของสถานีไฟฟ้าย่อยก่อนที่ จะหาสถานีไฟฟ้าอื่นมาช่วยรับโหลดคือสถานีนี้ต้องมี Connected Load ต่ออยู่หรือก็คือค่า Status of Connected Feeder (Sti) เป็นค่า On อยู่ด้วย ถ้าในกรณีที่ไม่ใช่ Connected Feeder ใด on อยู่จะไม่สามารถทำการถ่ายคลาดได้มีเฉพาะ Radial Load ซึ่ง Rad ไม่สามารถโอนถ่ายโหลดได้เช่นกัน
- ในการที่จะทำการ Switching จะต้องทำเนืองถึง Connected Load ด้วย

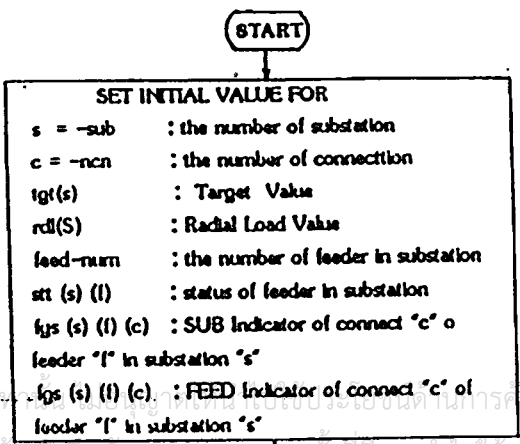
การทำงานของโปรแกรม

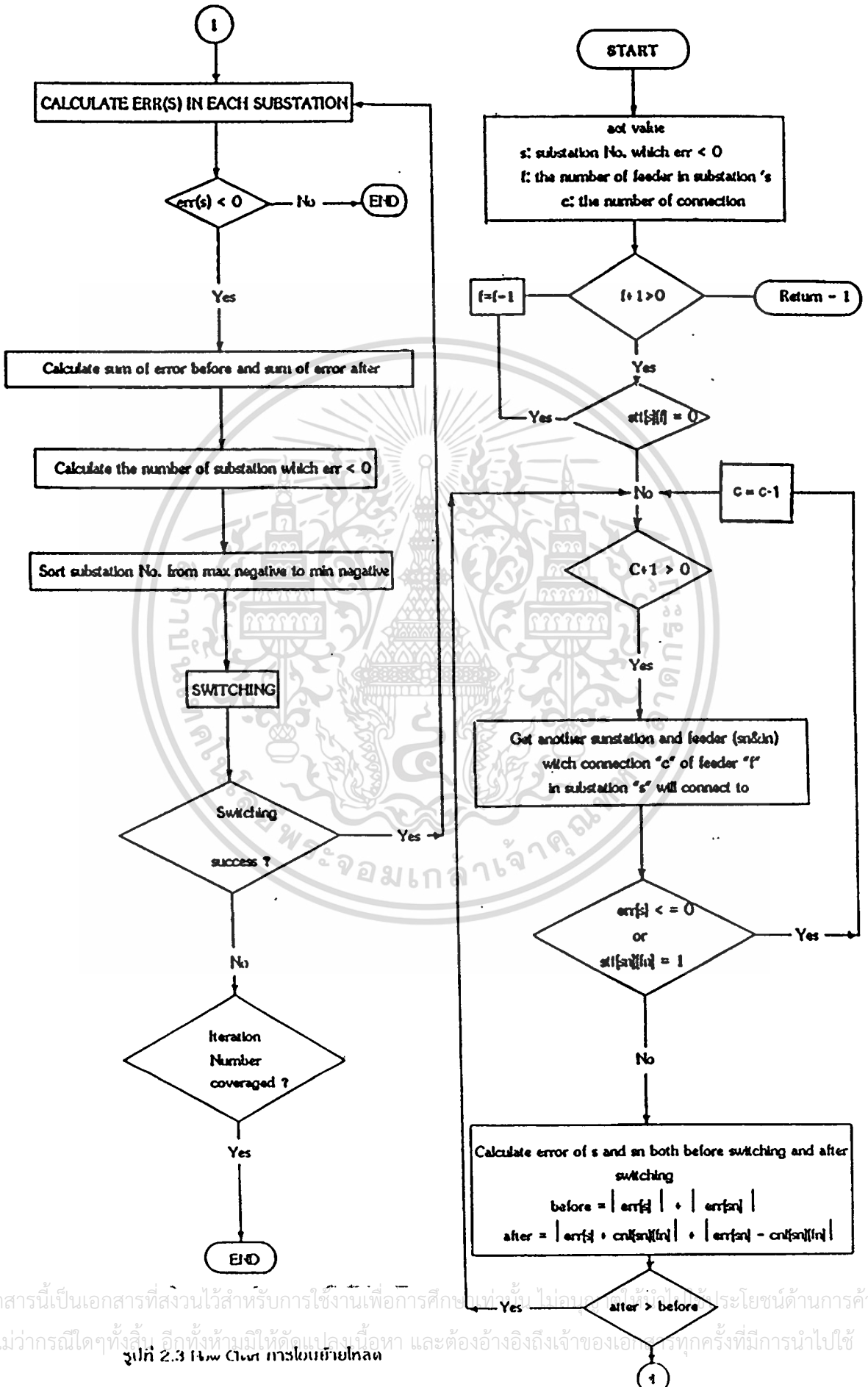
เป็นการทำการโอนย้ายโหลดโดยให้ค่าผลรวมของค่าสัมบูรณ์ของค่าผิดพลาด (Err) มีค่าน้อยของหลักใหญ่ก็คือโอนถ่ายโหลด (เฉพาะ Connected Load) จากสถานีที่มีค่า Err ต่ำ ไปสถานีที่มีค่า Err ที่มากพอ และขึ้นอยู่กับค่าของ Connected Load (Cn) ตามที่กล่าวมาแล้ว นอกจากนั้นต้องมีการตรวจสอบว่าสถานีอื่นที่ทำการถ่ายโหลดไปให้ต้องมีสถานะเป็น on หรือตรวจสอบได้ว่าสายสถานะเป็น off หรือตรวจสอบว่าสายจะทำการถ่ายโหลด และที่สำคัญที่สุดจะต้องแน่ใจว่า Switching แล้วจะสามารถลดค่า Sum of Error ได้แน่จึงจะทำการ Switching และก่อนที่จะเลือกสถานีใดมาทำการ Switching หรือโอนย้ายโหลด จะต้องเรียงลำดับให้สถานีที่มีค่า Err มากกว่าศูนย์มา ๆ มีสิทธิ์ได้ทำก่อน เพราะว่า จะทำให้ หม้อแปลงในสถานีไฟฟ้าย่อยซึ่งทำงานหนักมากยิ่งขึ้น เป็นอัตราค่าที่ก่อให้เกิดเสียหายต่ออายุการใช้งานและสิ้นเปลืองเพราะ Efficiency ก็จะตกลง

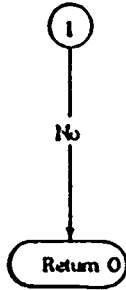
อนัน จึงต้องทำการโอนย้ายโหลดจากค่าที่ติดลบมาก ๆ ก่อนเมื่อทำการโอนย้ายเสร็จแล้วมาทำการแก้ไขข้อมูลต่าง ๆ ดังนี้คือ

- ค่า Act : Actual Value
- ค่า Err : Error Value
- ค่า Sum of Error : Sum of Error
- ค่า Sti : Status of Feeder

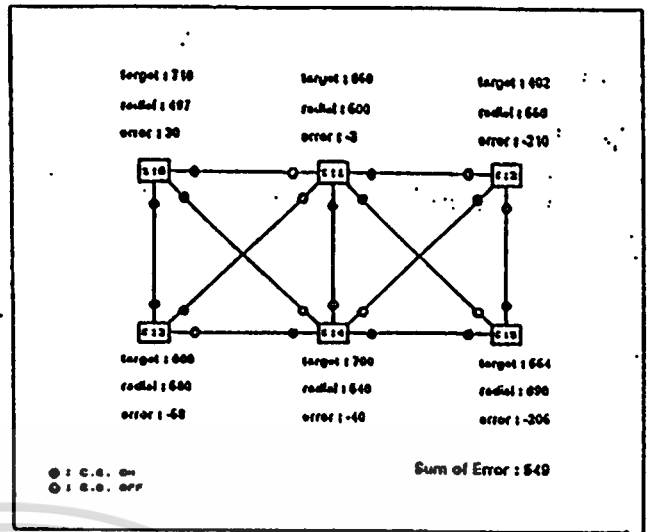
Flow Chart ของโปรแกรม





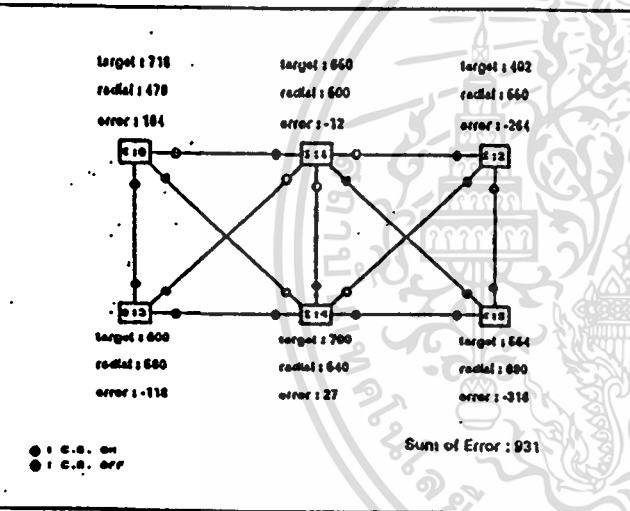


รูปที่ 2.4 Flow Chart ของการทำการ Switching



รูปที่ 3.3 รูปแสดงระบบของข้อมูลหลังการ Switching

3. ผลที่ได้จากการวิจัย



รูปที่ 3.1 รูปแสดงระบบข้อมูลก่อนการ Switching

SUB	sp	sd1	sd2	sd3	fl	c1	c2	c3	c4	c5	c6	c7	c8	c9	c10	c11	c12	c13	c14	c15	c16	c17	c18
0	710	650	470	30	1	00	1	-4	1	104													
1	650	600	500	-8	0	100	0	30	1	05	1	60	1	64									
2	402	610	650	210	0	64	1	64	0	78													
3	600	640	600	-60	0	00	1	70	0	80													
4	700	700	640	-60	1	64	0	45	0	41	0	63	1	100									
5	654	703	690	200	0	100	0	64	1	72													

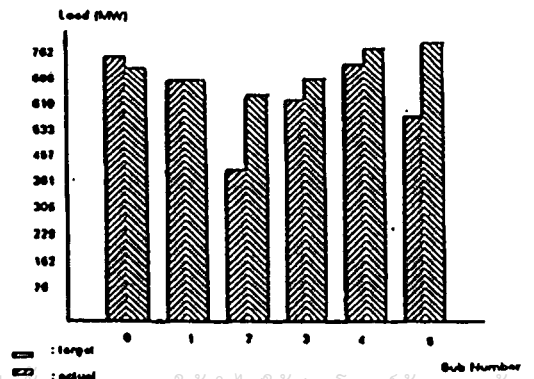
Sum of Error 540

รูปที่ 3.4 รูปแสดงระบบข้อมูลหลังการ Switching

SUB	sp	sd1	sd2	sd3	fl	c1	c2	c3	c4	c5	c6	c7	c8	c9	c10	c11	c12	c13	c14	c15	c16	c17	c18
0	710	650	470	30	0	00	1	05	0	104													
1	650	600	500	-8	1	100	0	30	0	05	1	60	1	64									
2	402	610	650	210	1	64	1	64	0	78													
3	600	640	600	-60	1	00	1	70	0	80													
4	700	700	640	-60	1	64	0	45	0	41	0	63	1	100									
5	654	703	690	200	1	100	0	64	1	72													

Sum of Error 931

รูปที่ 3.2 รูปแสดงตารางข้อมูลก่อนการ Switching

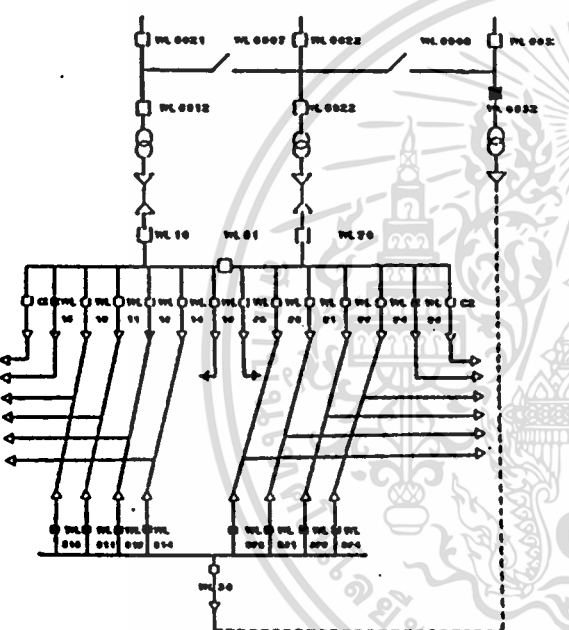


รูปที่ 3.5 รูปแสดงกราฟข้อมูลหลังการ Switching

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นิยมนำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4. บทสรุป

การโอนถ่ายโหลดในพื้นที่ย่อยๆ มีจำนวนสถานีไฟฟ้าย่อยในพื้นที่ไม่มากนักในแต่ละสถานีย่อยสามารถโอนถ่ายโหลดให้แก่กันได้ แต่ในกรณีที่มีการถ่ายโอนระหว่างพื้นที่ที่ต้องใช้ PC ควบคุม 1 ชุด เพื่อที่จะทำให้เกิดประสิทธิภาพสูงสุด โดยจะต้องมีหน่วยประมวลผลรวมเป็นตัวควบคุมและรับรู้ข้อมูลในแต่ละพื้นที่เพื่อให้หน่วยประมวลผลรวมนี้ เป็นตัวสั่งการโอนถ่ายโหลดระหว่างพื้นที่อีกทีหนึ่ง ซึ่งการประมวลผลรวมในแต่ละพื้นที่สามารถให้การ Switching ได้อย่างมีประสิทธิภาพ จำเป็นต้องอาศัยระบบฐานข้อมูลที่ใหญ่มาก และ Software นี้ควรจะเน้นทางด้านการทำงานจุดที่เกิดที่โอเวอร์โหลด และ จุดที่สามารถโอนถ่ายโหลดได้อย่างรวดเร็ว และ แอมป์เพื่อทำการ Switching ที่มีประสิทธิภาพสูงขึ้นดังรูปที่ 4.1



รูปที่ 4.1 แสดงลักษณะการถ่าย Load อัตโนมัติ จากรูปที่ 4.1 เป็นการแสดงการโอนถ่าย Load ในลักษณะอัตโนมัติ ซึ่งส่วนที่ต่ออยู่กับหม้อแปลง, protection relay ที่จะทำงานก็คือ differential relay และ tripping relay ซึ่งจะสั่ง Trip circuit breaker หน้าหลังหม้อแปลง เพื่อตัด fault ออกจากระบบการจ่ายไฟหลังจาก tripping relay ทำงาน และ incoming leader(bank) trip ก็จะมีสัญญาณส่งหม้อแปลง bay 3 และ circuit breaker 12 KV bay 3 ด้านที่ขนาด bay ที่เกิดเหตุเพื่อจ่ายไฟทันที ซึ่งเราสามารถอธิบายการเกิด Fault ในที่ต่าง ๆ ได้ดังนี้

4.1 กรณีที่เกิด fault ที่หม้อแปลง bay1 differential และ tripping relay จะทำการปลด WL-6912 และ WL-10 หลังจากนั้นจะสั่ง WL-6932 และ WL-30 เข้ามาและจะสั่ง WL-311, WL-312, WL313, WL314 โดยไม่ปลด WL-11, WL-12, WL-14 ออก เนื่องจาก fault ถูกกำจัดออกไปแล้ว(ดังรูปที่ 4.1) ส่วนของการทำงานวงจรอัตโนมัติของ bay 2 ก็จะเหมือนกับกับ bay 1

4.2 กรณีที่เกิด fault ที่ busbar 12 KV bay 1 WL-10, WL-01 จะ trip ด้วย over current หรือ earth fault relay จากนั้นวงจรอัตโนมัติจะทำงานโดยปลด circuit breaker (leader) ทางด้าน bay1 ออกหลังจากนั้นจะสั่ง WL-6932 และ WL-30 กับที่ตามด้วย

จะปลด WL-11, WL-12, WL-13, WL-14 และสั่ง WL-311, WL-313, WL314 ต่อที่จะสั่ง circuit breaker ทางด้าน bay 3 วงจรดังกล่าวจะ check ว่า circuit breaker ทางด้าน bay 1 ปลดจะไม่มี relay show เหนือเมื่อวงจรทำงานและ trip WL-11 เมื่อ WL-11 trip และ WL-11 ไม่มี relay show วงจรจะสั่ง close WL-311ทันที

4.3 กรณีเกิด fault ที่สายป้อนและ circuit breaker ทางด้าน bay 1 trip ไม่ออก over current หรือ earth fault relay ของ WL-10 และ WL-01 งาน trip WL-10 และ WL-01 วงจรก็จะทำงานเช่นเดียวกับกรณีเกิด fault ที่ busbar และจะไม่มีการ close circuit breaker ตัวผู้ที่ยานกับ circuit breaker ทางด้าน bay 1 ที่เกิดเหตุ เช่นกรณีเกิด fault ในสายป้อน WL-14 และ WL-14 trip ไม่ออกเนื่องจากมีเหตุขัดข้อง over current หรือ earth fault relay ของ WL-10, WL-01 ทำงานและปลด WL-10, WL-01 จากนั้นวงจรจะ trip WL-11, WL-12, WL-13, WL-14 ออกและจะสั่ง WL-311, WL-312, WL-313 เท่านั้น ส่วน WL-314 จะไม่มีคำสั่งสับเพื่อกำจัด fault ออกจากระบบ

4.4 กรณีเกิด fault ทาง busbar ทางด้าน bay 3 over current หรือ earth fault relay ของ WL-30 จะทำงานและ trip WL-30 ขณะเดียวกัน relay จะส่งสัญญาณไป off วงจรดังกล่าวเสีย เพื่อป้องกันการผิดพลาดที่จะตามมา

จากการทำงานดังกล่าวมานั้นจะใช้เวลาไปถึง 1 วินาที และถ้าวงจรดังกล่าวทำงานจะมีหลอด indicator show ที่แผง remote 12 KV bay 3 วงจรจะมี timer ตั้งไว้ 1 วินาทีนับจากวงจรเริ่มทำงาน เมื่อครบ 1 วินาที timer จะ block วงจร เพื่อไม่ให้ทำงานซ้ำและใบการ on-off วงจรอัตโนมัติเราสามารถ control ทั้งแผง remote control 12 KV bay 3 และจากทาง scada นอกจากนี้แต่ละสายป้อนยังมี switch on/off ที่จะเลือกใช้การทำงานแบบอัตโนมัติลักษณะเดียวกับ under frequency relay

5. เอกสารอ้างอิง

-Modern Power Station Practise Volume
4. The Control Electricity Generation Box. , London, 1964, Chapter 22 หน้า 121-123
-Carr, T.H. Sub-Station Practise Second Edition 1952, London หน้า 446-447-438-442, 346
-American Electricians Handbook by Terrall Croft. หน้า 346-366
-New Delopment in Co-ordinate Control & Protection G.Koch, H.Menzel, Stemen ag, Nu5ramberg, Germany
-IEEE Transtion on Power Delivery, Vol.4, No.2, April 1989 "Loss Minitzation of Distribution Feeder : Optimality and Algorithm", หน้า 1281-1287

6. กิติกรรมประกาศ

ขอขอบคุณ ผศ.มณฑล ฉลาจินดาโรภกุล อาจารย์ที่ปรึกษาซึ่งได้ให้แนวทางในบทความวิจัยนี้ให้สำเร็จลุล่วงไปด้วย

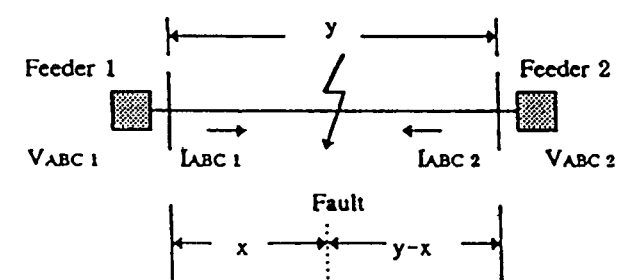
ด้วยวิธีการส่งสัญญาณพัลส์ออกไปแล้ววัดสัญญาณที่ตอบสนองกลับมาและประมวลผลออกทางจอภาพของคอมพิวเตอร์ ซึ่งการประมวลผลนั้นจะมีการตรวจวัดระดับแรงดันและกระแสปกติ, กระแสขณะเกิดฟอลต์และหลังการเกิดฟอลต์ที่ส่งมาจาก RTU (Remote Terminal Unit) บันทึกและรายงานผลทางเครื่องพิมพ์เก็บเป็นข้อมูลทางสถิติเพื่อนำมาตั้งสมมุติฐานเพื่อใช้ในการคำนวณหาตำแหน่งฟอลต์ด้วยโปรแกรมต่อไป

2. ทฤษฎีและหลักการพื้นฐานในระบบการจัดการ

ทฤษฎีและหลักการพื้นฐานที่ใช้ในการหาตำแหน่งฟอลต์ในสายส่งนั้นมีอยู่ด้วยกันหลายวิธี ซึ่งแต่ละวิธีก็มีวิธีการที่พิจารณาแตกต่างกันออกไป ดังนี้

- 2.1 วิธีการพิจารณาขนาดอิมพีแดนซ์ของโหลดเพื่อคำนวณหาตำแหน่งแรงดันและกระแสที่จุดเกิดฟอลต์นั้น[4]
- 2.2 วิธีการพิจารณาขนาดอิมพีแดนซ์ของแหล่งจ่ายทั้ง 2 ด้านเพื่อเขียนสมการแรงดันจากวงจรสมมูลย์ของสายส่งขณะที่เกิดฟอลต์[5]
- 2.3 วิธีการพิจารณาค่าอิมพีแดนซ์ปรากฏขณะที่เกิดฟอลต์ โดยใช้อัตราส่วนของแรงดันและกระแสที่เลือกซึ่งขึ้นอยู่กับชนิดของฟอลต์ โดยค่าอิมพีแดนซ์นี้จะแปรผันกับระยะทางที่เกิดฟอลต์[6]
- 2.4 วิธีการพิจารณาค่าความต้านทานฟอลต์ โดยใช้ค่าแรงดันและกระแสที่ได้จากการวัด โดยตั้งสมมุติฐานว่าค่ารีแอกแตนซ์ของฟอลต์มีค่าอิมพีแดนซ์เท่ากับศูนย์เพื่อหาค่าฟอลต์ออกไป[7]

ซึ่งในบทความนี้จะนำเสนอการหาตำแหน่งฟอลต์ในสายส่งแบบสองจุดป้อนโดยใช้วิธีการวัดค่าแรงดันและกระแสทั้ง 3 เฟสเมื่อมีการเกิดฟอลต์ในระยะต่าง ๆ เพื่อทำการคำนวณและวิเคราะห์ในรูปสถิติโดยใช้ข้อมูลของแรงดันและกระแสทั้ง 2 ด้านในระบบจำหน่ายทั้ง 3 เฟสที่ถูกบันทึกไว้ ดังวงจรสมมูลย์ของสายส่งขณะเกิดฟอลต์ ดังรูปที่ 1



จากรูปที่ 1 เมื่อมีฟอลต์เกิดขึ้นในระยะ x จาก Feeder 1 สามารถเขียนสมการเวกเตอร์ของแรงดันทั้ง 3 เฟสจาก Feeder ทั้ง 2 ได้ดังนี้

$$V_{ABC1} = VF_{ABC} + xZ_{ABC}I_{ABC1} \quad \dots(1)$$

$$V_{ABC2} = VF_{ABC} + (y - x)Z_{ABC}I_{ABC2} \quad \dots(2)$$

เมื่อกำหนดให้

- V_{ABC} = เวกเตอร์ของแรงดันในเฟส A,B,C
- I_{ABC} = เวกเตอร์ของกระแสในเฟส A,B,C
- VF_{ABC} = เวกเตอร์แรงดัน 3 เฟสที่จุดเกิด Fault
- Z_{ABC} = เมตริกซ์อิมพีแดนซ์ของสายส่ง/หน่วยความยาว
- y = ความยาวของสายส่งระหว่าง Feeder 1 และ Feeder 2
- x = ระยะการเกิด Fault จาก Feeder 1

และจัดให้อยู่ในรูปของเมตริกซ์ได้เป็น

$$\begin{bmatrix} Y_A \\ Y_B \\ Y_C \end{bmatrix} = \begin{bmatrix} M_A \\ M_B \\ M_C \end{bmatrix} [x] \quad \dots(3)$$

หรือ $Y = Mx \quad \dots(4)$

โดยที่

$$Y_j = V_{j1} - V_{j2} + y \sum_{i=A,B,C} Z_{ji} I_{i2}$$

$$M_j = \sum_{i=A,B,C} Z_{ji}(I_{i1} + I_{i2})$$

เมื่อ $j = A, B, C$

และจากสมการที่ (4) คูณทั้ง 2 ข้างด้วย M^* จะได้

รูปที่ 1 แสดงวงจรสมมูลย์ของสายส่งขณะเกิดฟอลต์ เพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

$$M^* Y = M^* Mx \quad \dots(5)$$

ไม่ว่ากรณีใดทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ดังนั้น ทำให้สามารถทราบระยะเวลาการเกิดฟอลต์ได้ โดยวิธี ลีส-สแควร์ (Least-Square) นั่นคือ

$$x = (M^* M^{-1}) M^* Y \quad \text{---(6)}$$

และจากที่กล่าวมาข้างต้น ถ้าสัญญาณของทั้ง 2 Feeder ไม่มีการซิงโครไนส์กัน แสดงว่า ระยะฟอลต์ที่คำนวณได้ผิดพลาดตามไปด้วยเนื่องจาก มุมต่างเฟสกัน ดังนั้น จากสมการที่ 2 จึงเขียนใหม่ได้เป็น

$$V_{ABC2}e^{j\theta} = VF_{ABC} + (y - x)Z_{ABC}I_{ABC2}e^{j\theta} \quad \text{---(7)}$$

โดยกำหนดให้

$V_{ABC2}e^{j\theta}, I_{ABC2}e^{j\theta}$ = ซิงโครไนส์เฟสเซอร์ของแรงดันและกระแสที่ feeder 2
ดังนั้น จากสมการที่ (1) และ (7) สามารถเขียนใหม่ได้เป็น

$$V_{ABC1} - V_{ABC2}e^{j\theta} = xZ_{ABC}I_{ABC1} - yZ_{ABC}I_{ABC2}e^{j\theta} + xZ_{ABC}I_{ABC2}e^{j\theta} \quad \text{---(8)}$$

จากสมการที่ (8) จัดรูปสมการให้อยู่ในรูปของตัวแปรที่ไม่ทราบค่า $(x, e^{j\theta})$ ได้เป็น

$$\frac{V_{j1}}{V_{j2}} = M1_j x + M2_j e^{j\theta} + M3_j + x e^{j\theta} \quad \text{---(9)}$$

เมื่อกำหนดให้

$$M1_j = \frac{1}{V_{j2}} \sum_{i=A,B,C} Z_{ji} I_{i1}$$

$$M2_j = 1 - \frac{y}{V_{j2}} \sum_{i=A,B,C} Z_{ji} I_{i2}$$

$$M3_j = \frac{1}{V_{j2}} \sum_{i=A,B,C} Z_{ji} I_{i2}$$

และเมื่อกำหนดให้ $x = x1$ และ $e^{j\theta} = x2$ แทนลงในสมการที่ (9) จะได้

$$Y_j = M1_j x1 + M2_j x2 + M3_j x1 x2 \quad \text{---(10)}$$

โดยที่ $Y_j = \frac{V_{j1}}{V_{j2}}$

และจากสมการที่ (9) และ (10) นั้น มีตัวแปร 2 ตัวที่ไม่ทราบค่า นั่นคือ ระยะ x ซึ่งเป็นค่าจริงและ $e^{j\theta}$ ซึ่งเป็นค่าเชิงเส้น ในการหาค่าตัวแปรทั้ง 2 สามารถทำได้โดยการกำหนดค่า $x1_0, x2_0$ แล้วจึงคำนวณหาค่า $x1$ และ $x2$ โดยวิธี ลีส-สแควร์ ได้เป็น

$$\begin{bmatrix} x1 \\ x2 \end{bmatrix}_k = \begin{bmatrix} x1 \\ x2 \end{bmatrix}_{k-1} + [H^* H]^{-1} H^* \Delta y \quad \text{---(11)}$$

เมื่อกำหนดให้

$$H = \text{Jacobian matrix} = \frac{dY_j}{dX_j}$$

H^* = คอนจูเกตทรานซ์โพสของ matrix H

$\Delta y = Y$ ที่วัดได้ - Y ที่คำนวณ $(k-1)$

Y ที่คำนวณ $(k-1)$ = ค่า Y ที่คำนวณจากสมการที่ (10) ที่ $k-1$

และจากสมการที่ (11) เป็นการคำนวณแบบวนรอบและปรับค่าของ $x1, x2$ ไปจนกว่าจะเข้าสู่ค่าคงที่ค่าหนึ่ง ซึ่งจะดีที่ระยะที่เกิดฟอลต์และมุมซิงโครไนส์นั่นเอง ซึ่งการหาค่าของ Jacobian matrix สามารถทำได้โดย

$$\frac{\partial Y_j}{\partial X1} = M1_j + M3_j x2_{k-1}$$

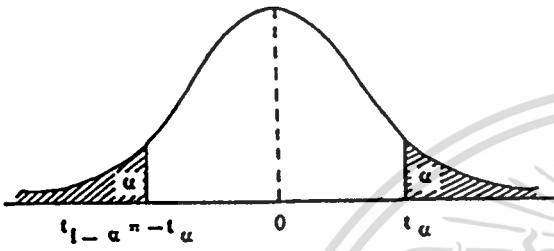
$$\frac{\partial Y_j}{\partial X2} = M2_j + M3_j x1_{k-1}$$

เมื่อ $j = A, B, C$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่สามารถนำออกเผยแพร่โดยไม่ได้รับอนุญาต

ไม่ว่ากรณีใดทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ฟอสและระยะที่เกิดฟอลท์จาก Remote terminal unit ส่งมายังศูนย์ควบคุมที่รายงานผลออกทางเครื่องพิมพ์(2)แล้วจึงนำข้อมูลมาวิเคราะห์ทางสถิติโดยใช้วิธีการแจกแจงแบบสุ่มปกติ(Normal Sampling Distribution)ที่เรียกว่า Student t-Test [3]เพื่อกำหนดสมมติฐานความน่าจะเป็นของ random sample ที่ตกอยู่ระหว่างค่า t ทั้งสองจะเท่ากับพื้นที่ใต้โค้งที่อยู่ระหว่าง ordinates ทั้งสองนั้น ดังรูปที่ 2



รูปที่ 2 แสดงการสมมาตรของการแจกแจงแบบ t

โดยทั่วไปจะกำหนดให้ค่า t มีค่าสูงกว่า α ซึ่ง α อาจจะถูกกำหนดให้เท่ากับ 0.1, 0.05, 0.0025, 0.001 หรือ 0.005 เมื่อ α แทนค่า t ที่อยู่เหนือพื้นที่เท่ากับ α เช่น ค่า t ที่มี $df = 10$ พื้นที่ที่เหลือจะเท่ากับ 0.025 ทางขวามือ นั่นคือ $\alpha = 0.025 = 2.228$ เนื่องจากการแจกแจงแบบสมมาตรและมีค่าเฉลี่ยเท่ากับศูนย์ ดังนั้น $t(1-\alpha) = -\alpha$

และจากข้อมูลการเกิดฟอลท์ที่ส่งมาจาก RTU ที่ติดตั้งบนเสาไฟฟ้า สามารถนำมาวิเคราะห์ข้อมูลทางสถิติเพื่อหาค่าต่าง ๆ ได้ดังนี้

$$S_m^2 = \frac{\sum x_i^2 - m \bar{x}^2}{m-1} \quad \dots(12)$$

$$S_n^2 = \frac{\sum y_j^2 - n \bar{y}^2}{n-1} \quad \dots(13)$$

$$S_p^2 = \frac{(m-1)S_m^2 + (n-1)S_n^2}{m+n-2} \quad \dots(14)$$

$$t = \frac{\bar{x} - \bar{y}}{\sqrt{S_p^2 \left[\frac{1}{m} + \frac{1}{n} \right]}} \quad \dots(15)$$

โดยที่กำหนดให้

$\sum x_i^2$ = ผลรวมของข้อมูลในระยะเวลาที่เกิดฟอลท์ ยกกำลังสองของ Up-steam

\bar{x} = ค่าเฉลี่ยของข้อมูลในระยะเวลาที่เกิดฟอลท์ ทั้งหมดของ Up-steam

\bar{x}^2 = ค่าเฉลี่ยของข้อมูลในระยะเวลาที่เกิดฟอลท์ ทั้งหมดของ Up-steam ยกกำลังสอง

m = จำนวนข้อมูลที่ส่งเข้ามาในระยะเวลาเกิดฟอลท์ ของ Up-steam

\bar{y} = ค่าเฉลี่ยของข้อมูลในระยะเวลาที่เกิดฟอลท์ ทั้งหมดของ Down-steam

\bar{y}^2 = ค่าเฉลี่ยของข้อมูลในระยะเวลาที่เกิดฟอลท์ ทั้งหมดของ Down-steam ยกกำลังสอง

n = จำนวนข้อมูลที่ส่งเข้ามาในระยะเวลาเกิดฟอลท์ของ Down-steam

S_m^2 = ค่าเฉลี่ยของกลุ่มข้อมูลที่เกิดฟอลท์ของ Down-steam

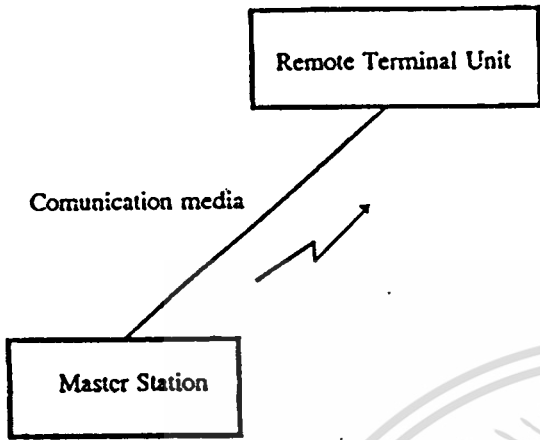
S_n^2 = ค่าเฉลี่ยของกลุ่มข้อมูลที่เกิดฟอลท์ของ Up-steam

S_p^2 = ค่าความแปรปรวนของทั้ง Up-steam กับ Down-steam

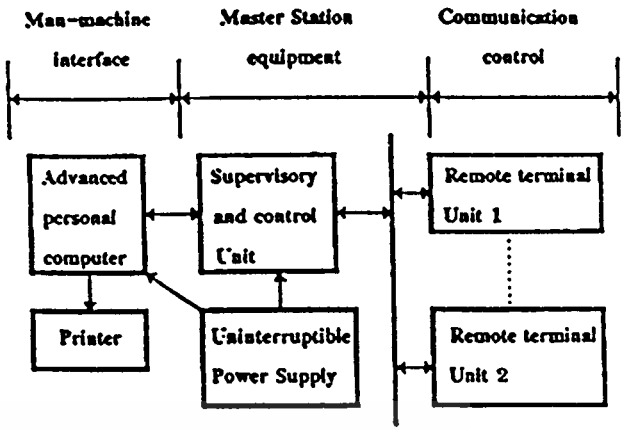
t = ค่าที่คำนวณได้เพื่อนำไปเปรียบเทียบกับ α และ $m+n-2$ ในตาราง t-Test

หลังจากคำนวณค่าทางสถิติได้แล้วก็พร้อมที่จะทำการตัดสินใจโดยนำไปเปิดตาราง t-Test เพื่อหาค่า α , $m+n-2$ เมื่อกำหนดให้ระดับความเชื่อมั่น (α) = 0.1 และ degree of freedom ($m+n-2$) = 46 จะได้ค่า $t = 1.301$ จากข้อกำหนดที่กล่าวไว้ว่าค่า t ที่คำนวณได้ต้องมากกว่า α จึงจะปฏิเสธ H_0 ซึ่งแสดงว่ามีฟอลท์เกิดขึ้นและแสดงว่าค่าสถิติหรือ sample value ตกอยู่ในพื้นที่วิกฤติ (critical region) และขณะเดียวกันก็ยอมรับ (H_a) แต่ถ้าค่าสถิติไม่ตกอยู่ในพื้นที่วิกฤติก็ยอมรับ (H_0) หรืออาจกล่าวได้ว่า fail to reject H_0 และการจัดการพลังงานในระบบจำหน่ายนั้นจะประกอบด้วยส่วนสำคัญ 3 ส่วนได้แก่ ศูนย์ควบคุม(Master Station),

ระบบสื่อสาร(Communication media)และสถานีระยะไกล (Remote terminal unit) ดังรูปที่ 3



รูปที่ 3 โครงสร้างของระบบ Distribution Automation System



รูปที่ 4 แสดงลักษณะการทำงานของ Master Station

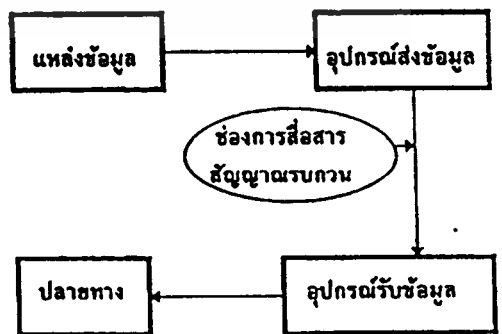
2.2 ระบบสื่อสาร(Communication media) การส่งถ่ายข้อมูลที่มีความหมายจากแหล่งข้อมูลหนึ่งไปสู่อีกแหล่งข้อมูลหนึ่ง สามารถแบ่งเป็น 2 ประเภทคือ การสื่อสารตามสายและการสื่อสารทางวิทยุสื่อสารจากทั้ง 2 ประเภทสามารถแบ่งเป็นส่วน ๆ ได้ดังนี้คือ(รูปที่ 4)

ส่วนที่ 1 แหล่งข้อมูลซึ่งผู้ควบคุมจะเป็นผู้จัดส่งและรับข้อมูลหรือส่งด้วยภาพ ฯลฯ

ส่วนที่ 2 อุปกรณ์ส่งข้อมูล จะทำหน้าที่เปลี่ยนข้อมูลจากแหล่งข้อมูลในรูปรหัสและส่งสัญญาณไปที่ช่องของการสื่อสารหรือช่องสัญญาณนั้น ๆ

ส่วนที่ 3 อุปกรณ์รับข้อมูล เมื่อมีการรับสัญญาณจากเครื่องส่งกำลังเครื่องรับสัญญาณจะมีกำลังอ่อนลงและยังมีสัญญาณรบกวนมาแทรกซึ่งเครื่องรับจะต้องมีความสามารถเลือกรับและขยายเอาเฉพาะสัญญาณความถี่ที่ต้องการ

ส่วนที่ 4 ปลายทาง เป็นอุปกรณ์ทางไฟฟ้าที่แปลงสัญญาณออกมาในลักษณะที่เป็นข่าวสาร (รูปที่ 5)



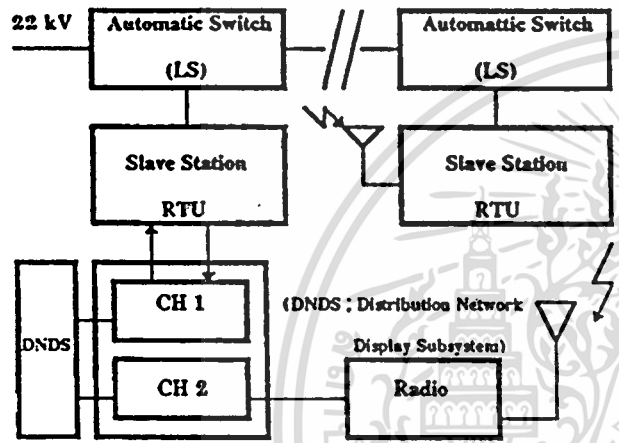
รูปที่ 5 ส่วนต่าง ๆ ของระบบสื่อสารเบื้องต้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นับญาติให้มาใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.3 สถานีระยะไกล(Remote Terminal Unit ; RTU)

เป็นอุปกรณ์ลูกข่ายที่ใช้ในการควบคุมการ เปิด - ปิด โหลด เบรกสวิตช์ (LS ; สวิตช์ตัดตอนแรงสูงที่ดับอาร์คด้วยแก๊ส SF6) ที่ติดตั้งอยู่ในระบบจำหน่ายโดยรับคำสั่งควบคุมการทำงานจากแม่ข่ายที่ติดตั้งอยู่ที่ศูนย์สั่งการ โดยแบ่งออกเป็น 2 ชนิด คือ ชนิดที่ควบคุมด้วยสายเคเบิล(cable type)และชนิดที่ควบคุมด้วยสัญญาณวิทยุสื่อสาร(Radio Type) ดังรูปที่ 6 และรูปที่ 7



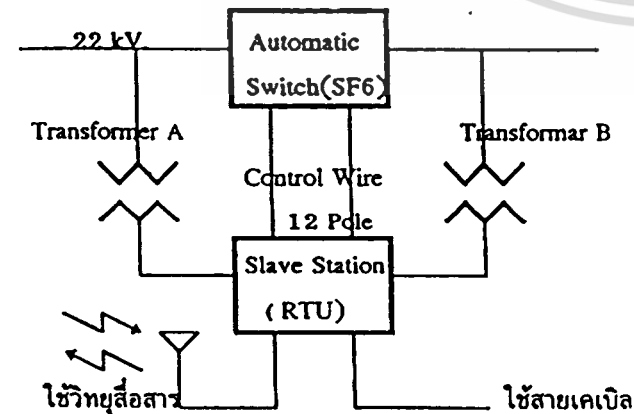
รูปที่ 6 แสดงลักษณะของ Remote Terminal Unit ทั้ง 2 ชนิด

3. โครงสร้างของระบบและการทำงาน

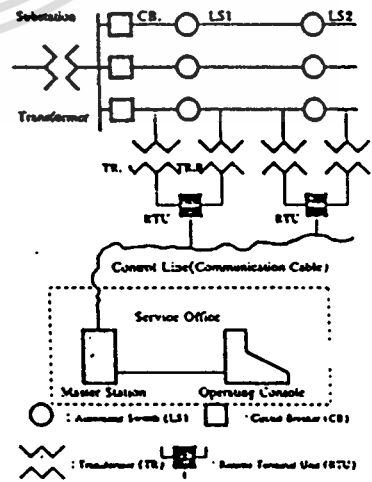
โครงสร้างของระบบการจัดการพลังงานในระบบจำหน่ายอัตโนมัติจะประกอบด้วยส่วนต่าง ๆ ดังตารางที่ 1 และรูปที่ 8

ส่วนตรวจสอบสถานะของอุปกรณ์ป้องกันและอ่านข้อมูลจากเครื่องวัดที่สถานีไฟฟ้า(Supervisory and Telemetry Sub-System)	ประกอบด้วยอุปกรณ์ - ชุดส่งข้อมูล - ชุดรับข้อมูล - แผงแสดงข้อมูล
ส่วนตรวจสอบสถานะและควบคุมอุปกรณ์ตัดตอนในระบบจำหน่าย (Telecontrol Sub-System)	ประกอบด้วยอุปกรณ์ - ชุดควบคุมแม่ข่าย - ชุดควบคุมลูกข่าย
ส่วนแสดงผลแผนผังระบบจำหน่ายและแผนควบคุม(Distribution Network Display Sub-System)	ประกอบด้วยอุปกรณ์ - หน่วยประมวลผลกลาง - แผงผังระบบจำหน่าย - แผงควบคุมระบบ
ส่วนจำลองสถานการณ์เพื่อการศึกษาอบรม(Dispatcher Training Simulate)	ประกอบด้วยอุปกรณ์ - ชุดคอมพิวเตอร์ - ส่วนแสดงผลแผนผังระบบจำหน่ายและแผนควบคุมระบบ
ส่วนของความต้านทานต่อลงดิน (Neutral Grounding Resistance)	ประกอบด้วยอุปกรณ์ - Grounding ของหม้อแปลงไฟฟ้า

ตารางที่ 1 แสดงส่วนประกอบของระบบการจัดการ



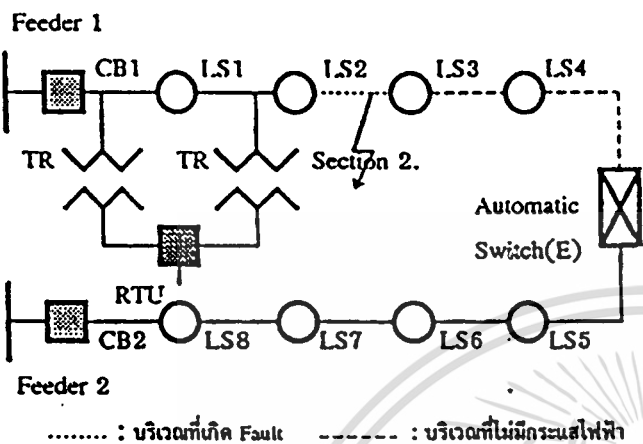
รูปที่ 7 แสดงส่วนประกอบของชุด Remote Terminal Unit



รูปที่ 8 แสดงโครงสร้างของระบบการจัดการ

ไม่ว่ากรณีใดก็ตาม อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ขั้นตอนการทำงานของระบบการจัดการจะใช้การตรวจสอบหาฟอลต์ที่เกิดขึ้นในระบบจำหน่ายซึ่งมีลักษณะการทำงานดังรูปที่ 9



รูปที่ 9 แสดงขั้นตอนการตรวจหาส่วนที่เกิดฟอลต์ในระบบจำหน่าย

จากรูป เมื่อมีฟอลต์เกิดขึ้นใน feeder 1 ที่ section 2. (ระหว่าง LS2 และ LS3) มีขั้นตอนการทำงานของระบบดังนี้
3.1 Circuit Breaker(CB.1) ที่ Substation จะตัดวงจรทำให้เกิดไฟดับในบริเวณ Feeder 1 ทั้งหมด

3.2 Circuit Breaker(CB.1) Reclosed ครั้งที่ 1 หลังจากเวลาผ่านไป 1 Sec. (LS2 และ LS3 ยังไม่เปิดวงจร)

3.3 เนื่องจากยังคงมีฟอลต์อยู่ที่ Circuit Breaker(CB.1) ตัดวงจรเป็นครั้งที่ 2 ทำให้ LS2 และ LS3 เปิดวงจร

3.4 เมื่อเวลาผ่านไป 20 Sec. Circuit Breaker(CB.1) Reclosed ครั้งที่ 2 ทำให้ LS2 ปิดวงจรหลังจากเวลาเกิน X-Time. ในกรณีนี้ยังคงมีฟอลต์ในระบบจำหน่ายใน Section 2. อยู่ ทำให้ Circuit Breaker(CB.1) ตัดวงจรเป็นครั้งที่ 3 LS2 เปิดวงจรและ Lock ตัวเองอยู่ในสภาวะเปิดวงจรทำให้เกิดไฟดับในสายป้อนที่ 1 ทั้งหมดอีกครั้ง

3.5 Circuit Breaker(CB.1) Reclosed ครั้งที่ 3 ในเวลา 20 Sec. ต่อมาในขณะที่ LS2 อยู่ในสภาวะเปิดวงจร

3.6 ในขณะนี้สามารถตัดสินใจได้ว่าจุดที่ทำให้เกิดฟอลต์อยู่ใน Section 2. แน่แน่นอน

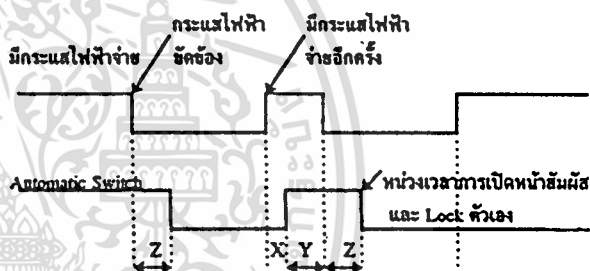
3.7 Automatic reclosing sectionlizer(E) ตัดตอนเชื่อมโยง จะถูกสั่งให้ปิดวงจรโดยสั่งการจากศูนย์ควบคุมซึ่งทำให้ระบบจำหน่ายใน Section 3. และ(LS4)มีกระแสไฟฟ้าใช้ตามปกติ

3.8 หลังจากค้นหาสาเหตุที่ทำให้เกิดฟอลต์และทำการ

ซ่อมแซมระบบจำหน่าย Section 2. เรียบร้อยแล้วทำการปิดวงจร LS2 ด้วยการควบคุมระยะไกลจากศูนย์ควบคุม ทำให้สายป้อน 1 และ 2 มีกระแสไฟฟ้าใช้ตามปกติ

4. ฟังก์ชันการหน่วงเวลา

จากรูปที่ 7 หม้อแปลง 2 ชุดที่ติดตั้งในแต่ละด้านของ Automatic reclosing sectionlizer(LS) ทำหน้าที่แปลงแรงดันไฟฟ้าจากแรงดันสูงให้เป็นแรงดันต่ำเพื่อจ่ายให้กับระบบควบคุมของ Remote Terminal Unit และ Automatic reclosing sectionlizer ในส่วนของ Automatic reclosing sectionlizer จะทำหน้าที่เปิดวงจรหลังจากเกินเวลา Z-time(2 Sec.)และจะทำหน้าที่ปิดวงจรภายในเวลา X-time(สามารถตั้งเวลาทำงานได้ระหว่าง 5-995 Sec.)หลังจากมีการจ่ายไฟกลับเข้ามาใหม่ ดังรูปที่ 10



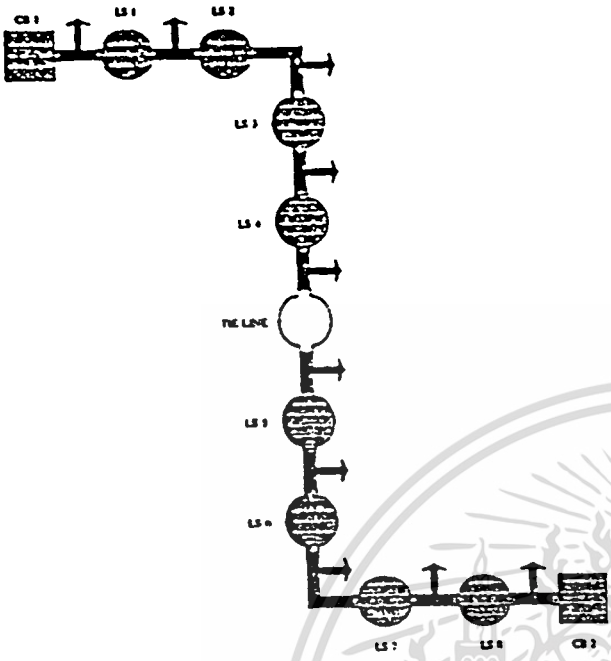
รูปที่ 10 แสดงขั้นตอนการทำงานของฟังก์ชันการหน่วงเวลา

5. การทดสอบและผลการทดสอบของระบบ

จากการทดสอบกับชุดจำลองระบบจำหน่าย ซึ่งมี Single line diagram ดังรูปที่ 11 ทดสอบโดยใช้ข้อมูลจากแบบจำลองระบบจำหน่ายด้วยโปรแกรมภาษา C โดยกำหนด Input Data ดังรูปที่ 12 และได้ผลการทดลองโดยโปรแกรมจะทำการคำนวณค่ากระแสในจุดที่เกิดฟอลต์โดยอัตโนมัติพร้อมทั้งแสดงแผนผังสถานที่เกิดฟอลต์โดยอัตโนมัติ บนหน้าจอคอมพิวเตอร์ซึ่งแบ่งเป็นฟอลต์ประเภทต่าง ๆ ได้ดังรูปที่ 13, รูปที่ 14 และรูปที่ 15

จากรูปที่ 12 เป็นฟอลต์ประเภท Three phase fault เกิดขึ้นระหว่าง LS1 และ LS2 ทำให้ Automatic reclosing sectionlizer 2 ตัวนี้ทำการตัดกระแสไฟฟ้าช่วง LS1 และ LS2 ออกทำให้ feeder 1 ที่จ่ายอยู่นั้นดับทั้งหมดซึ่งทำให้ส่วนที่ตรวจสอบสถานะและควบคุมอุปกรณ์และส่วนที่แสดงแผนผังระบบจำหน่ายแสดงผลบนหน้าจอคอมพิวเตอร์ทำให้ผู้ควบคุมทราบว่าเกิดฟอลต์ขึ้นจึงต้องทำการ Tie line เพื่อจ่ายแรง

ต้นไฟฟ้าจาก feeder 2 ไปยังบริเวณที่ไม่เกิดฟอลท์แทน ทําให้บริเวณนั้นมีกระแสไฟฟ้าใช้ตามปกติ

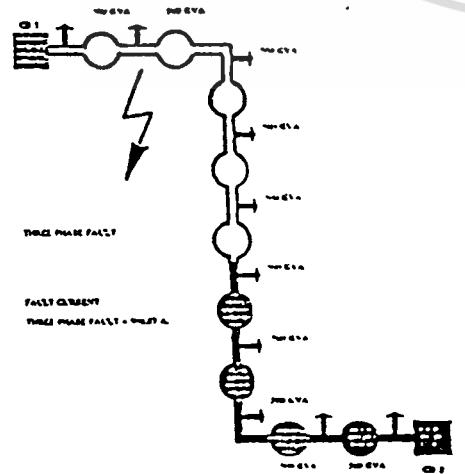


รูปที่ 11 แสดง Single line diagram จากชุดจำลอง

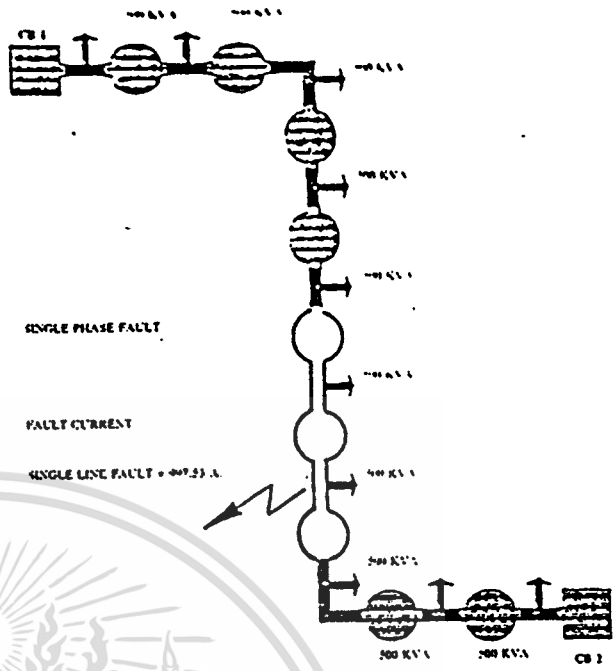
INPUT DATA	
SYSTEM VOLTAGE :	2400 (KV)
CAPACITY OF EGAT TRANSFORMER :	2400
FAULT AT BUS 1 - 3 :	1000 (A)
FAULT AT BUS 1 - 3 :	1000 (A)

[Enter] to move [Esc] to quit

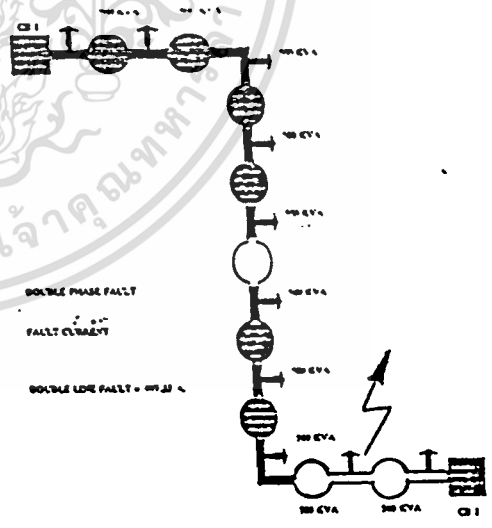
รูปที่ 12 แสดงการป้อนข้อมูลให้กับโปรแกรมภาษา C



รูปที่ 13 แสดงการเกิด Three phase fault



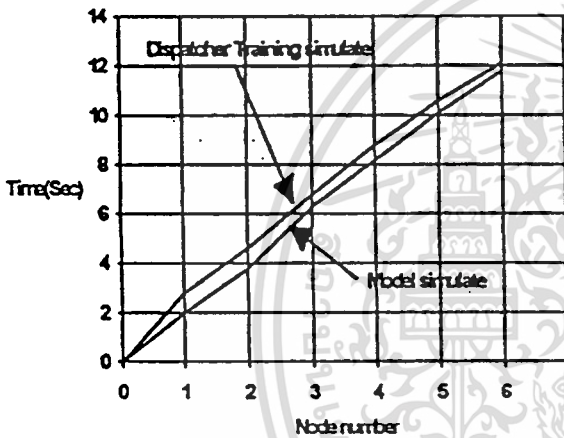
รูปที่ 14 แสดงการเกิด Single phase fault จากรูปที่ 14 เมื่อเกิดฟอลท์ประเภท Single phase fault ขึ้นระหว่าง LS5 และ LS6 ทําให้ Automatic reclosing sectionlizer 2 ตัวนี้ทำการตัดกระแสไฟฟ้าข้าม LS5 และ LS6 ออกเป็นผลทําให้ไฟดับในช่วงดังกล่าว



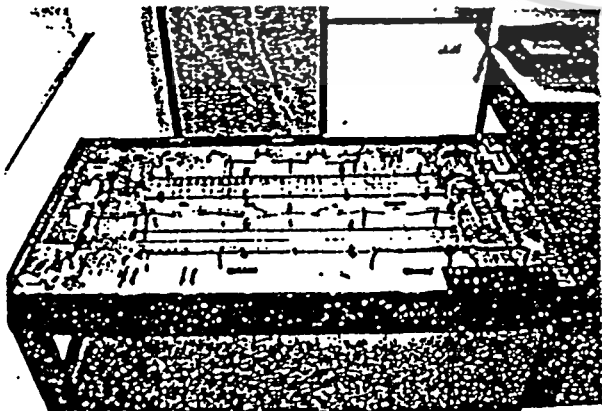
รูปที่ 15 แสดงการเกิด Double phase fault จากรูปที่ 15 เมื่อเกิดฟอลท์ประเภท Double phase fault ขึ้นระหว่าง LS7 และ LS8 ทําให้สถานีไฟฟ้าย่อย 2 ที่ feeder 2 ไม่สามารถจ่ายกระแสไฟฟ้าให้ LS5 และ LS6 ได้ จึงทําให้ต้องทําการ Tie line เพื่อจ่ายแรงดันไฟฟ้าจาก feeder 1 ไปยังบริเวณที่ไม่เกิดฟอลท์แทนทําให้บริเวณ LS5 และ LS6 มีกระแสไฟฟ้าใช้ตามปกติ

Node number	Interval (km)	Conductor code	Installed (kVA)
1	5	1	500
2	5	2	500
3	5	3	500
4	5	4	500
5	5	5	500

ตารางที่ 2 จุดต่อใน section ต่าง ๆ



รูปที่ 16 กราฟแสดงการทำงานของ Automatic reclosing sectionlizer ในเวลาต่าง ๆ



จากรูปที่ 16 แสดงการทำงานของ Automatic reclosing sectionlizer ในช่วงเวลาที่เกิดฟอลท์ ขึ้นในระบบที่ Node number 1-5 ของทั้ง 2 feeder ดัง Single line diagram ของรูปที่ .11 ทำใหทราบทว่าในช่วงเวลาที่เกิดฟอลท์ในระบบ Automatic reclosing sectionlizer จะทำการเปิดวงจรในช่วงเวลาเพียง 2 sec โดยประมาณ ที่ Node number 1 และที่ Node number 5 ใช้เวลามากกว่า 11.8 sec ผลเนื่องจากระยะทางและการทำงานของ Automatic reclosing sectionlizer ที่ใช้เวลาในการส่งข้อมูลมาที่ศูนย์ควบคุม ซึ่งการนำ RTU(Remoted terminal unit)มาช่วยในการควบคุม Automatic reclosing sectionlizer นี้จะทำให้ใช้เวลาในการตรวจสอบฟอลท์ของระบบจำหน่ายลดลงกว่าระบบเดิมที่มีอยู่ ที่ต้องใช้พนักงานค้นหากจุดที่เกิดฟอลท์เองและต้องใช้เวลาในการแก้ไขมากทำให้เกิดผลกระทบต่อกราดำยโหลดในระบบจำหน่าย

6. บทสรุป

การจัดการพลังงานในระบบจำหน่ายโดยอัตโนมัติ ในบทความนี้ ได้ทำการทดสอบโดยสร้างชุดจำลองระบบจำหน่าย (Model simulate)และนำคอมพิวเตอร์เข้ามามีส่วนร่วมในการตรวจจับการเกิดฟอลท์ซึ่งรับสัญญาณจากอุปกรณ์ตรวจจับที่เป็นตัวตรวจจับการเปลี่ยนแปลงทางกายภาพแล้วนำผลที่ได้ นั้นแสดงบนจอภาพของคอมพิวเตอร์เป็น Single line diagram ของระบบจำหน่ายจากทั้ง 2 feeder ในการทดสอบนี้ ได้จำลองภาคส่วนของระยะทางโดยให้ทุก ๆ 5 กม.ติดตั้ง Remote terminal unit และ Automatic reclosing sectionlizer อย่างละ 1 ตัวโดยใช้ Relay แทนอุปกรณ์ทั้งสอง ดังรูปที่ 11 และ รูปที่ 17 เมื่อนำโปรแกรมไปทดสอบกับชุดจำลองที่สร้างขึ้น (Model simulate) และนำโปรแกรมดังกล่าวไปทดสอบกับชุดจำลองของศูนย์ควบคุมต้นแบบ (2) (Dispatcher Training Simulate) จะสังเกตได้ว่าระยะฟอลท์ที่ได้จากชุดจำลองที่สร้างขึ้นน้อยกว่าชุดจำลองของศูนย์ควบคุมต้นแบบ เพียงแต่ชุดจำลองของศูนย์ควบคุมต้นแบบเป็นการทดสอบและวัดสัญญาณที่เกิดขึ้นจริง ซึ่งยังมีค่าตัวแปรที่ไม่สามารถควบคุมและสั่งการได้อีกมาก จึงทำให้ผลที่ได้จากทั้ง 2 ชุด มีค่าผิดพลาดไม่เท่ากัน แต่อย่างไรก็ตามจากผลการทดสอบของชุดจำลองทั้ง 2 ก็ยังแสดงให้เห็นว่ามีแนวโน้มนำไปใช้ในงานควบคุมระบบจำหน่ายในอนาคตได้จริง

เอกสาร รูปที่ 17 แสดงแผนผัง Model simulate เพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

7.เอกสารอ้างอิง

- [1] เอกสารการฝึกอบรม องค์การความร่วมมือระหว่างประเทศของญี่ปุ่น การไฟฟ้าส่วนภูมิภาค
- [2] ศูนย์ปฏิบัติการจ่ายไฟฟ้าย่อยนคร 1 และ นคร 2 การไฟฟ้าส่วนภูมิภาค
- [3] A.C.M. Chen, "Automated Power Distribution" IEEE Spectrum, Vol.19, No.4, pp.55-60, April 1982.
- [4] Aoki, Node Distribution line carrier System for section switch Telecontrol 1986 IEEE WM 229-9
- [5] Kato, Naito kohno, kanawa, Shoji Computer based Distribution automation IEEE, 1935 Power Industry computer Application Conference page 374-380
- [6] T. J. Kendrew, P. A. Schnieper, "Distributed Microprocessor Application to Electric Utility Distribution Protection and Control System", presented at the American control conference, San Diego, California, June 6-8, 1984.
- [7] R. Balakrishnan and A. Pahwa, "A Computer assisted intelligent storm outage evaluator for power distribution systems", IEEE Transactions on Power Delivery, pp.1591-1597, July 1990.
- [8] R. D Christie, H. Zadehgo, and M. M. Habib, "High impedance fault detection in low voltage networks", IEEE Transactions on Power Delivery, pp. 1829-1834, Oct 1993.
- [9] "Status of high impedance fault detection", IEEE Transactions on Power Apparatus and Systems, pp.638-643, Mar 1985.
- [10] S. J. Balser, K. A. Clements and D. J. Lawrence, "Amicroprocessor based technique for detection of high impedance faults", IEEE Transactions on Power Delivery, pp. 252-258, July 1986.
- [11] Srinivasan, K., St-Jacques, A. A "New Fault Location Algorithm for Radial Transmission Lines with Loads" IEEE Transactions on Power Delivery, July 1989, pp.1676-1682.
- [12] Takagi, T., Yamakoshi, Y., Yamaura, M., Kondow, R., Matsushima, T. "Development of a New Type Fault Locator using the One-Terminal Voltage and Current Data." IEEE Transactions on Power Apparatus and System, August 1982, pp.2892-2898.
- [13] W. H. Kersting, IEEE Distribution Planning Working Group Report, "Radial Distribution Test Feeders", IEEE Transactions on Power Systems, pp. 975-985, Aug 1991.
- [14] Y. Hsu and et al, "An expert system for locating distribution system faults", IEEE Transactiona on power Delivery, pp.366-372, Jan 1991.

ประวัติผู้เขียน

- ชื่อ** นายรัชชัย ธรรมประสิทธิ์
- เกิดวันที่** 18 มีนาคม พ.ศ. 2510 จังหวัด นครสวรรค์
- การศึกษา** อดสาหกรรมศาสตรบัณฑิต สาขาวิศวกรรมไฟฟ้า
มหาวิทยาลัยเอเซียอาคเนย์ ปีการศึกษา พ.ศ. 2536
- ผลงานทางวิชาการ** 1. “การจัดการพลังงานในสถานี่ไฟฟ้าย่อย” การประชุมวิชาการ
ทางวิศวกรรมไฟฟ้า ครั้งที่ 17 (2537)
2. “การจัดการพลังงานในระบบจำหน่ายโดยอัตโนมัติ”
วารสารวิศวกรรมลาดกระบัง ปีที่ 14 ฉบับที่ 1 (เมษายน 2541)



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้