

Ladder สำหรับงาน Sequence control

May 2023

Mitsubishi Electric Factory Automation (Thailand)

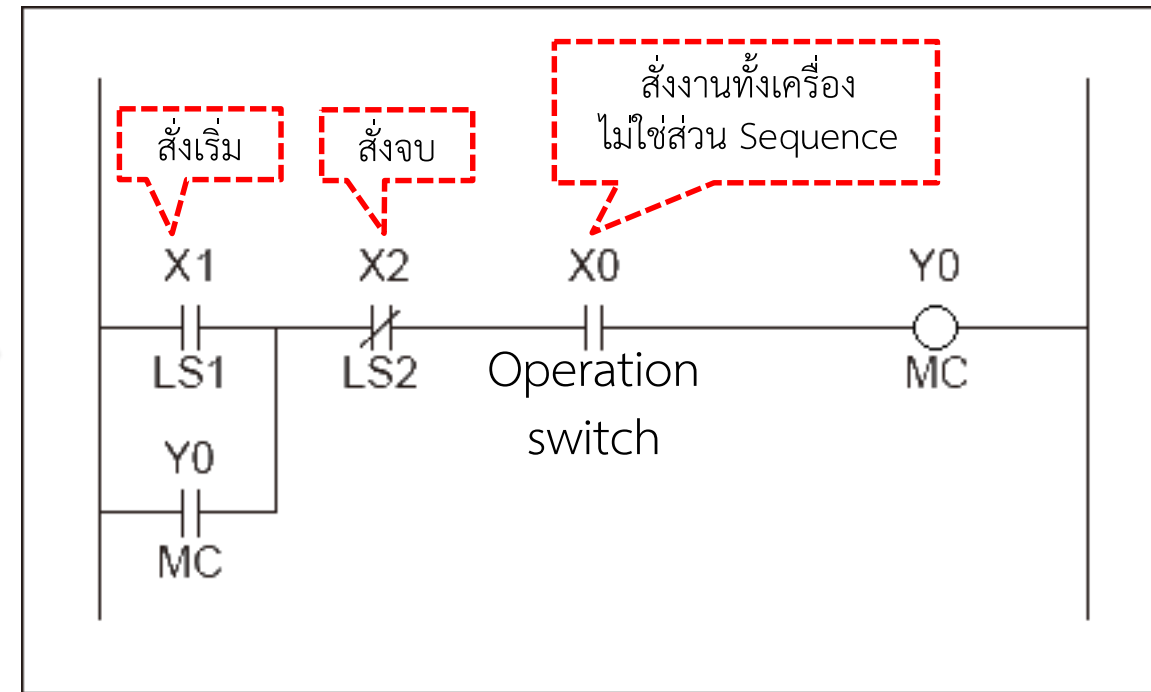
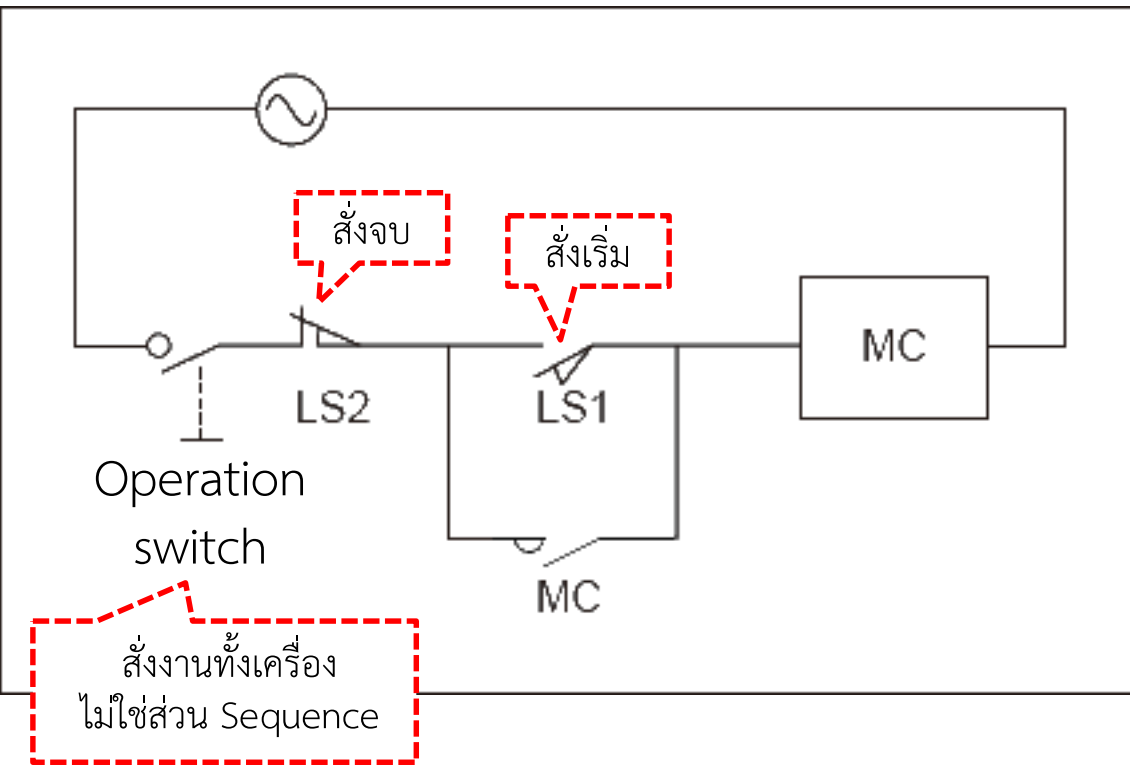
- Training iQ-F Basic (GX Works3).pdf (บทที่ 5)
- อ่านเอกสารนี้และคู่มือด้านบน ทดลองด้วย GX Works3 และใช้ Simulation แทน PLC จริงได้
- Miroslaw Wcislik. (2003). Programming of Sequential System in Ladder Diagram Language,
สืบค้นเมื่อ 10/04/2023 จาก
https://www.researchgate.net/publication/317053114_Programming_of_Sequential_System_in_Ladder_Diagram_Language

FX5U devices and device numbers (initial parameter)

	Item	Symbol	Type	Notation	Point	Range	Latch (1)
Device (high speed)	Input	X	Bit	OCT	1024	0 to 1777	-
	Output	Y	Bit	OCT	1024	0 to 1777	-
	Internal Relay	M	Bit	DEC	7680	0 to 7679	500 to 7679
	Link Relay	B	Bit	HEX	256	0 to FF	No Setting
	Link Special Relay	SB	Bit	HEX	512	0 to 1FF	-
	Annunciator	F	Bit	DEC	128	0 to 127	No Setting
	Step Relay	S	Bit	DEC	4096	0 to 4095	500 to 4095
	Timer	T	Word	DEC	512	0 to 511	No Setting
	Retentive Timer	ST	Word	DEC	16	0 to 15	0 to 15
	Counter	C	Word	DEC	256	0 to 255	100 to 199
	Long Counter	LC	Double Word	DEC	64	0 to 63	20 to 63
	Data Register	D	Word	DEC	8000	0 to 7999	200 to 7999
	Latch Relay	L	Word	DEC	7680	0 to 7679	-
Device (standard)	File Register	R	Word	DEC	32768	0 to 32767	No Setting
	Link Register	W	Word	HEX	512	0 to 1FF	No Setting
	Link Special Register	SW	Word	HEX	512	0 to 1FF	-

- Programmable (Logic) Controller โปรแกรมเมเบิล (ลอจิก) คอนโทรลเลอร์ หรือ PLC
 - จากมาตรฐาน IEC 61131-1 Programmable controllers – Part 1: General information
 - Programmable (Logic) Controller (PLC) ระบบอิเล็กทรอนิกส์ที่ทำงานแบบดิจิทัล ออกแบบมาเพื่อใช้ในสภาพแวดล้อมทางอุตสาหกรรม ซึ่งใช้หน่วยความจำที่ตั้งโปรแกรมได้สำหรับการจัดเก็บข้อมูลภายในของคำสั่งสำหรับการนำฟังก์ชันเฉพาะไปใช้ เช่น ตรรกะ (ลอจิก) ลำดับ ตั้งเวลา การนับและคำนวณ เพื่อควบคุมเครื่องจักรหรือกระบวนการประเภทต่าง ๆ ผ่านอินพุตและเอาต์พุตแบบดิจิทัลหรือแอนะล็อก ทั้ง PLC และอุปกรณ์ต่อพ่วงที่เกี่ยวข้องได้รับการออกแบบเพื่อให้สามารถรวมเข้ากับระบบควบคุมอุตสาหกรรมได้ง่ายและใช้งานง่ายในทุกฟังก์ชันที่ต้องการ
- คนญี่ปุ่นเรียก PLC ว่า Sequence Controller หรือ Sequencer เพราะงานหลักของ PLC คือ Sequence Control หรือการควบคุมตามลำดับ
 - Sequence หมายถึงขั้นตอนการดำเนินงานอย่างต่อเนื่องตามกฎและข้อบังคับ
 - Sequence Control หมายถึงการควบคุมที่ดำเนินงานในแต่ละขั้นตอนตามเงื่อนไขและลำดับที่กำหนดไว้ล่วงหน้า
- การทดสอบมาตรฐานฝีมือแรงงานแห่งชาติ สาขาอาชีพช่างไฟฟ้า อิเล็กทรอนิกส์และคอมพิวเตอร์
สาขาช่างควบคุมด้วยระบบโปรแกรมเมเบิลลอจิกคอนโทรลเลอร์ (Programmable Logic Controller : PLC) ของไทย
ตรงกับ National Trade Skill Test: Electric equipment assembling (Sequence control) ของญี่ปุ่น

วงจร Self hold ในงาน Sequence control

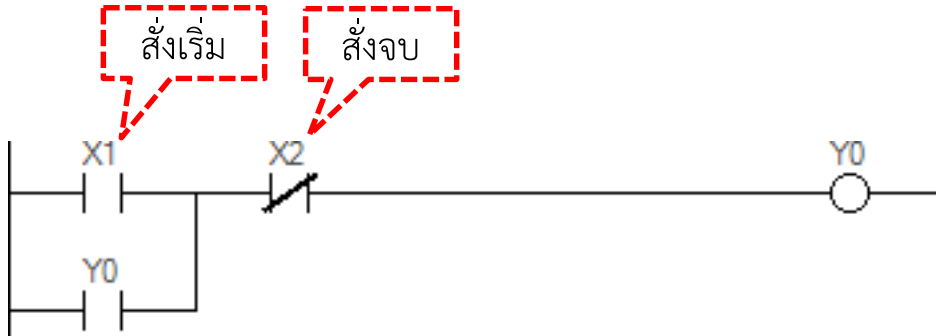


โปรแกรม PLC ที่นิยมมากกว่าการเขียนตามแบบไฟฟ้า เพราะ

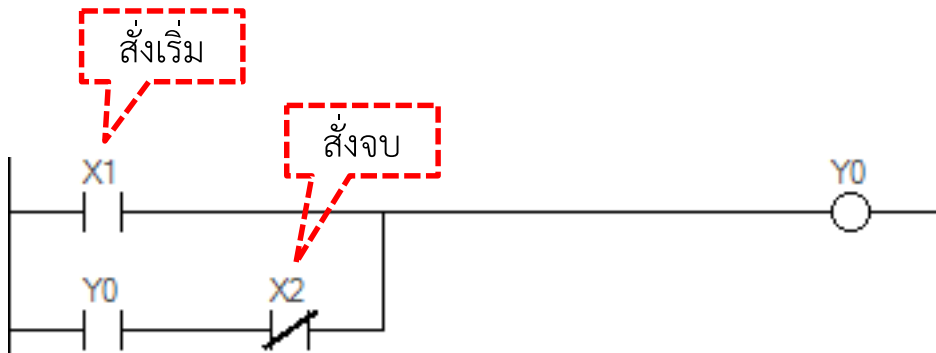
- โปรแกรมด้านบนใช้จำนวน Instruction code น้อยกว่า
- ผู้ใช้ PLC ยุคแรกต้องเขียน Instruction code แทน Ladder และใช้ PLC ที่หน่วยความจำน้อย จึงต้องใช้ Code สั้นที่สุด ซึ่งไม่จำเป็นในปัจจุบันที่ต้องการให้เข้าใจโปรแกรมง่ายที่สุด

โปรแกรม Self hold ที่ควรใช้

- ถ้ามีเฉพาะส่วน Self hold ที่ควบคุม Sequence

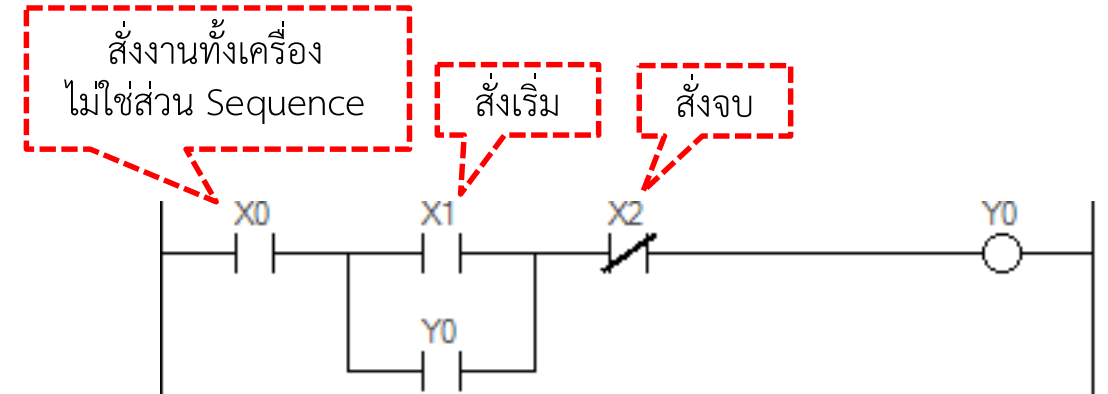


ให้ลำดับความสำคัญของหน้าสัมผัสสั่งจบสูงกว่า
สั่งเริ่มไม่ได้ถ้าหน้าสัมผัสสั่งจบตัดวงจรอยู่

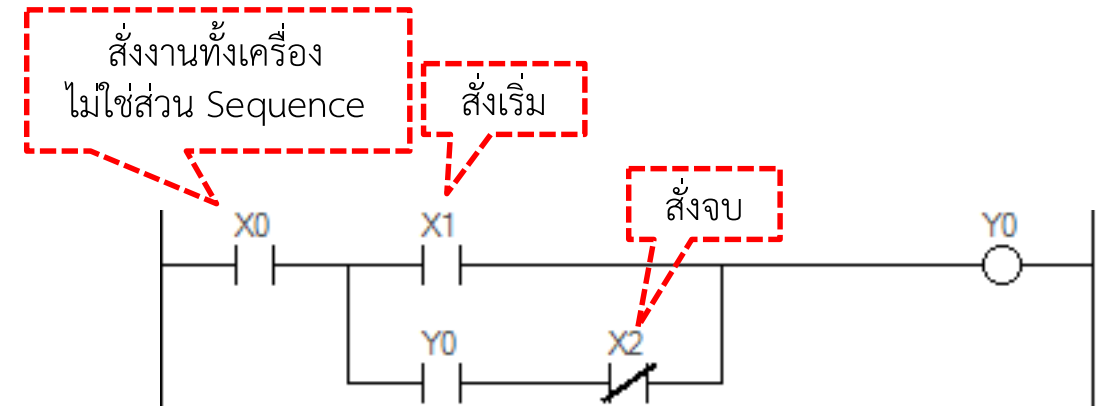


ให้ลำดับความสำคัญของหน้าสัมผัสสั่งเริ่มสูงกว่า
สั่งเริ่มได้แม้หน้าสัมผัสสั่งจบตัดวงจรอยู่

- ด้านซ้ายสุดเขียนส่วนสั่งงานทั้งเครื่องที่ไม่ใช่ส่วน Sequence



ให้ลำดับความสำคัญของหน้าสัมผัสสั่งจบสูงกว่า



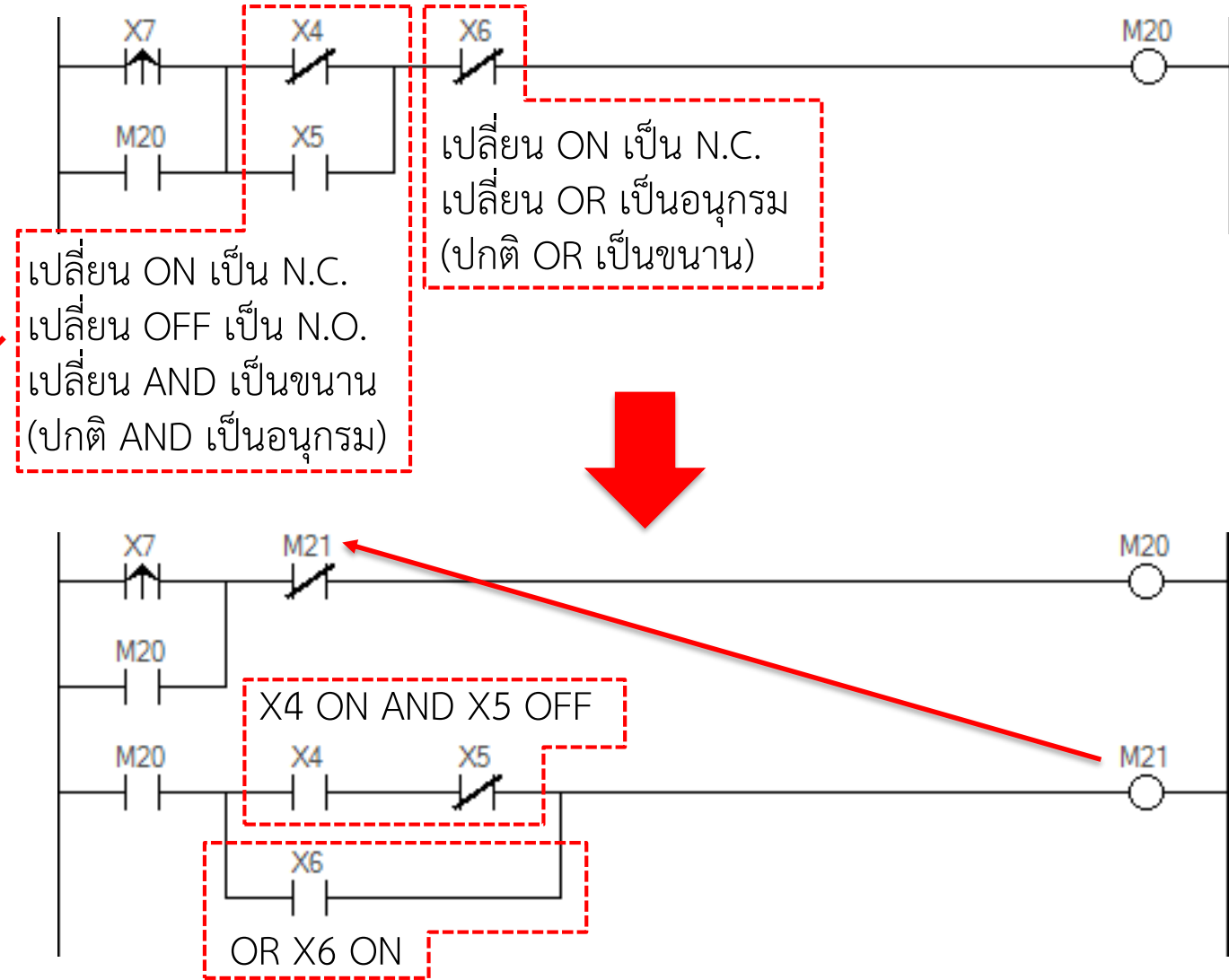
ให้ลำดับความสำคัญของหน้าสัมผัสสั่งเริ่มสูงกว่า

หน้าสัมผัสสั่งจบที่เงื่อนไขซับซ้อน

- จบเมื่อ X4 ON และ X5 OFF หรือ X6 ON
= X4 ON AND X5 OFF OR X6 ON
ทำ AND ก่อน OR เหมือนทำ * ก่อน +

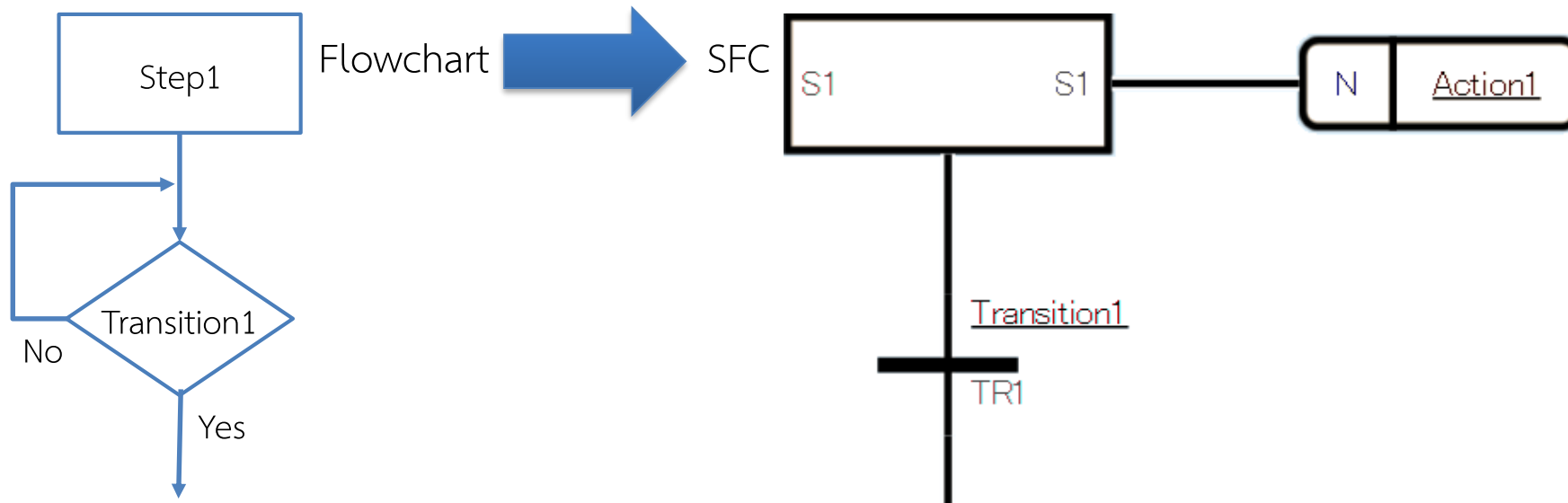
เขียนหน้าสัมผัสตัดการทำงานของ Self hold จะเข้าใจยาก

- เพิ่มโปรแกรมรวมหน้าสัมผัส Self hold และหน้าสัมผัสสั่งจบ
- เขียนโปรแกรมตามเงื่อนไขโดยตรง เข้าใจง่าย
- ใช้หน้าสัมผัส N.O. ของ Coil ที่ได้จากโปรแกรมไปสั่งจบ



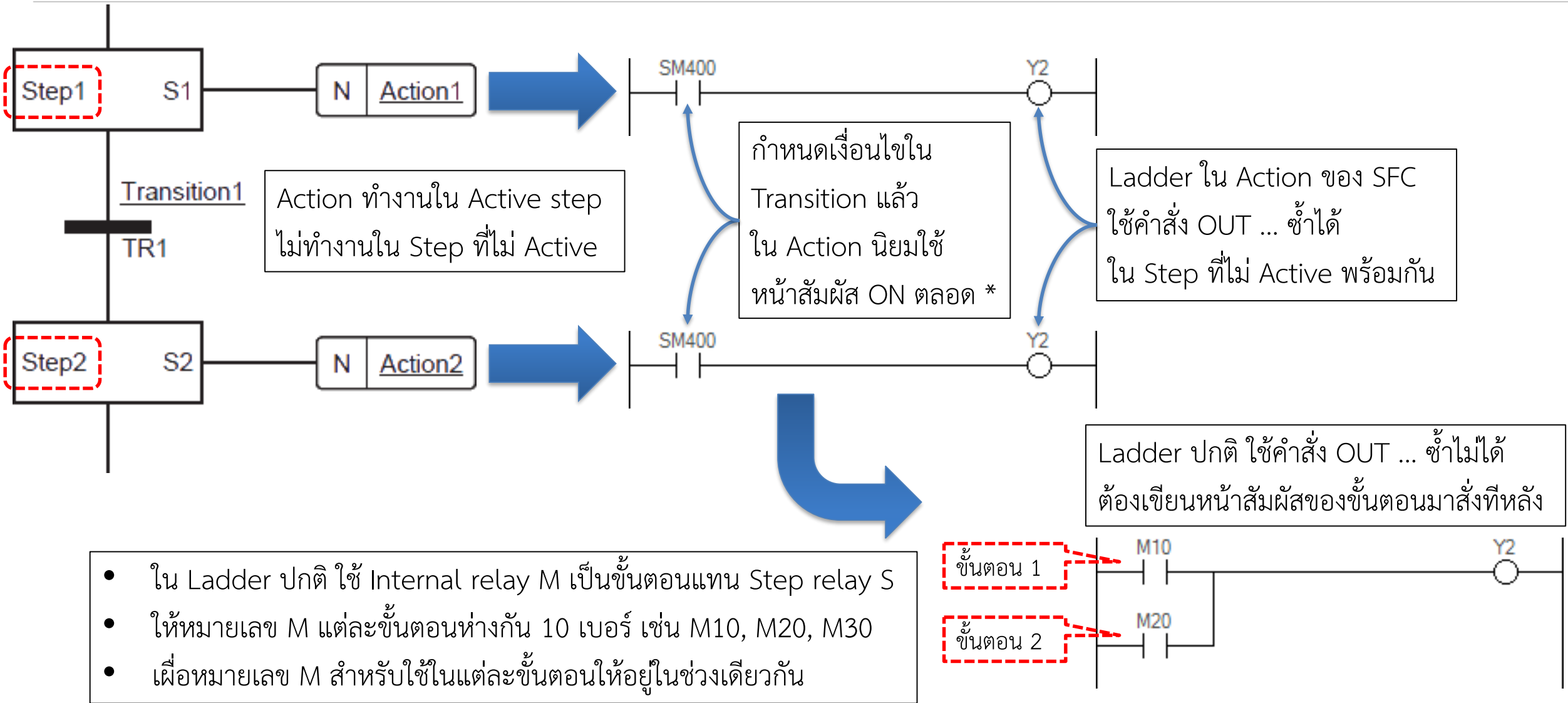
โปรแกรม Ladder เลียนแบบภาษา SFC

- ภาษา SFC (Sequential Function Chart) เป็นภาษา PLC ชนิดหนึ่งในมาตรฐาน IEC 61131-3 ที่เหมาะกับการ Sequence control
- ดัดแปลงจาก Flowchart เป็น Chart การควบคุมตามลำดับ (Sequence control) ภายในมีภาษา PLC อื่น (นิยมใช้ Ladder) เพื่อ
 - การทำงาน (Action) ในแต่ละขั้นตอน (Step), Step ที่ทำงานเรียกว่า Active step; Step เป็น SFC ภายในเขียน Action เป็น Ladder
 - เงื่อนไข (Condition) ของการเปลี่ยนขั้นตอน (Transition) ; Transition เป็น SFC ภายในเขียน Condition เป็น Ladder



- PLC รุ่น FX5U ใช้ภาษา SFC ได้ในรุ่นที่ Serial number ขึ้นด้วย 17X ขึ้นไป และ Update firmware เป็น Version 1.220 ขึ้นไป
- ใช้กับ GX Works3 version 1.75 ขึ้นไป โดย Version 1.76 ขึ้นไป Simulate การทำงานในภาษา SFC ได้
- การอบรมนี้จะสอนให้เขียน Ladder ปกติ (ไม่ใช่ภาษา SFC จริง) ให้ทำ Sequence control เลียนแบบการทำงานของ SFC

Ladder ใน Action ของ SFC กับ Ladder ปกติ

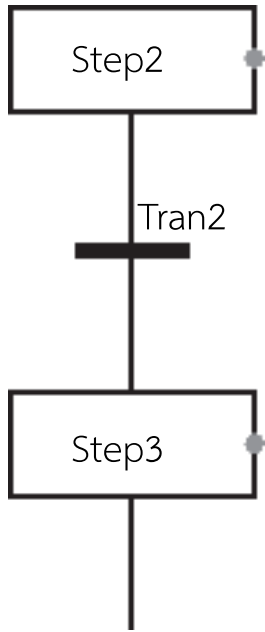


* SM400 Always ON เป็น Special relay กลุ่ม System clock ดูรายละเอียดจากคู่มือบท Appendix 1.2 Special Relay List (SM)

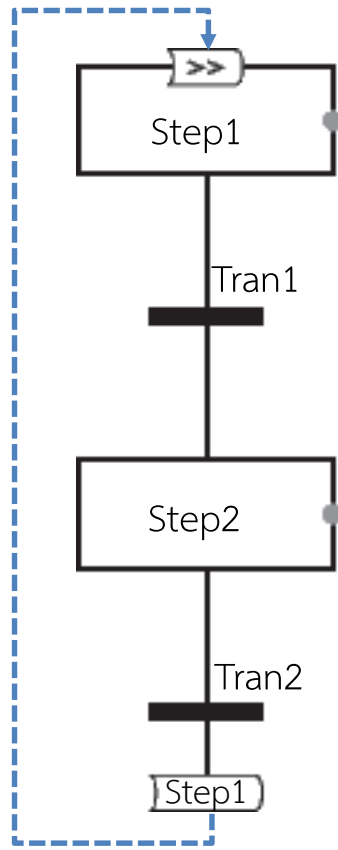
Transition แบบต่าง ๆ ของ SFC กับ Ladder ปกติ

ถ้า Active ที่ Step2
ก่อน Transition

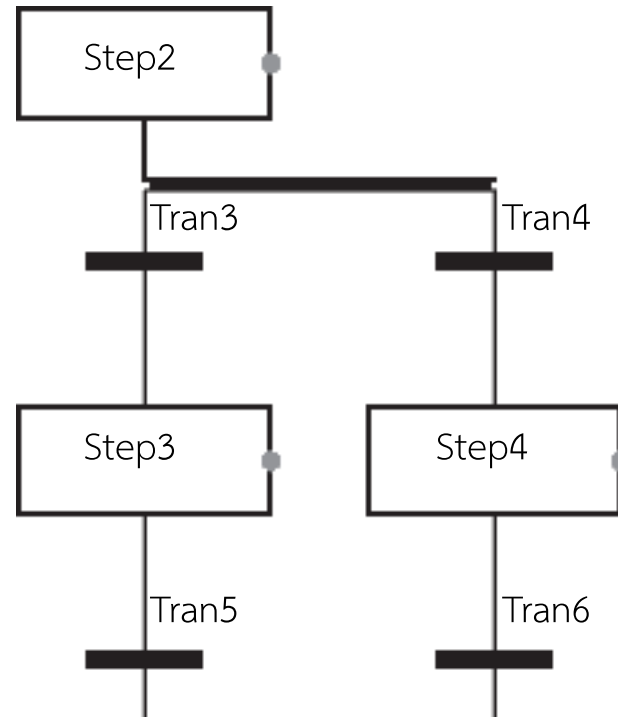
Series sequence
ต่อ Step ที่ตามมา
Tran2 จริง ทำ Step3



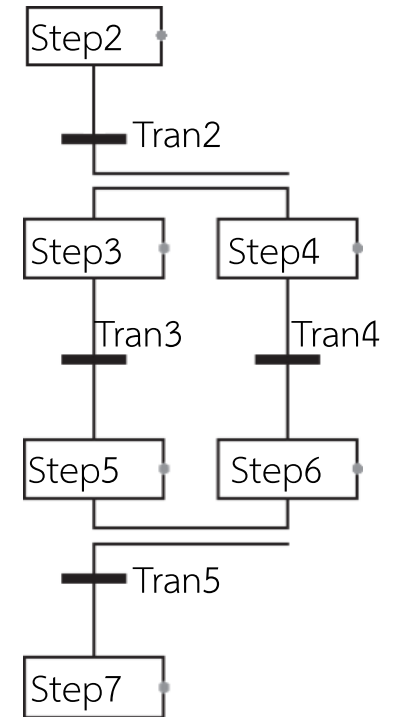
Jump sequence
กระโดดไปต่อ Step ที่ระบุไว้
Tran2 จริง ทำ Step1



Selective sequence
เลือกต่อ Step เดียว (ซ้ายสำคัญกว่า)
Tran3 จริง ทำ Step3
Tran3 ไม่จริง Tran4 จริง ทำ Step4

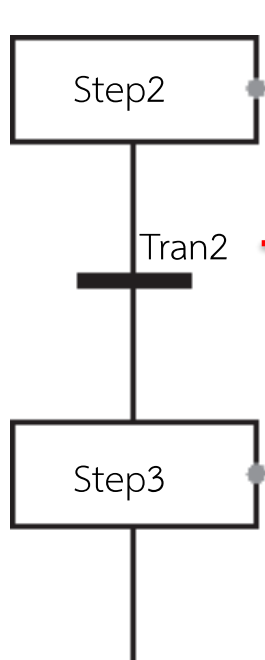


Simultaneous sequence
ต่อทุก Step ถัดมาพร้อมกัน
Tran2 จริง ทำ Step3 พร้อม Step4
ถ้ากำลังทำทั้ง Step5 และ Step6
แล้ว Tran5 จริง ทำ Step7

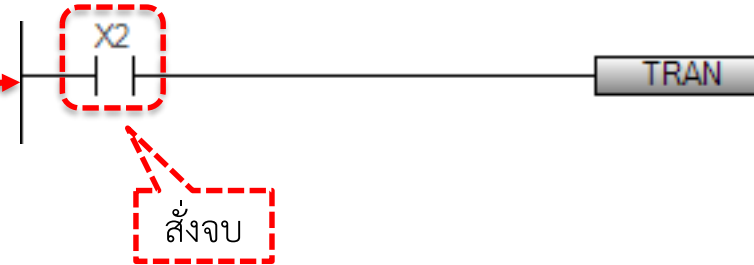


Series sequence เป็น Ladder ปกติ

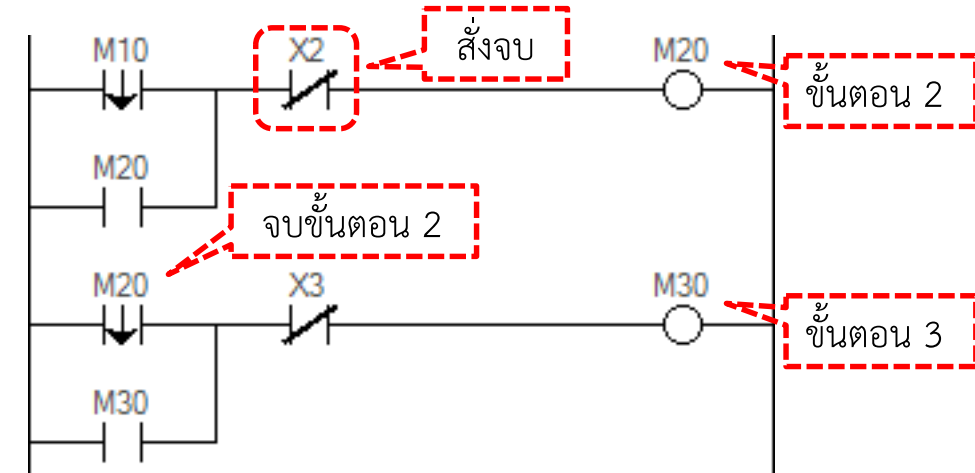
ต่อ Step ที่ตามมา
Tran2 จึง ทำ Step3



Ladder ใน Transition ทุกแบบของ SFC
ใช้หน้าสัมผัสสั่ง TRAN ให้ Step ใหม่ Active แทน
ดู Step ใหม่ตาม Chart ภาษา SFC

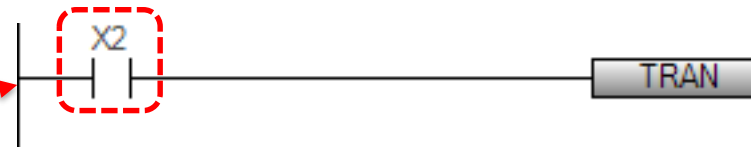
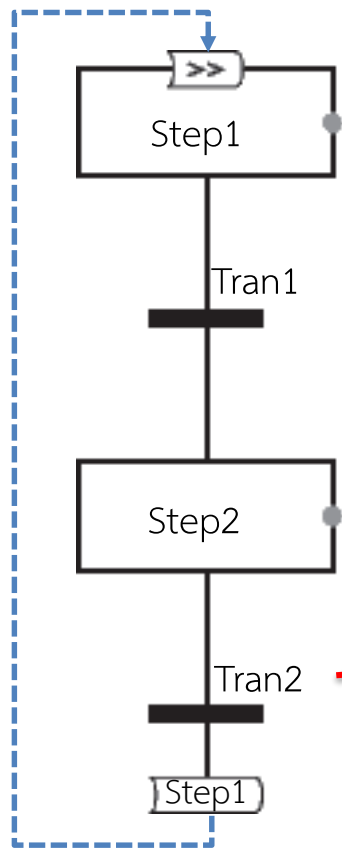


Ladder ปกติ ใช้หน้าสัมผัสตัดวงจร Self hold
ใช้ Pulse ขอบขาลงของขั้นตอนเดิมสั่งเริ่มขั้นตอนใหม่
(Pulse ขอบขาลง ON ใน Scan ถัดจากจบขั้นตอน)



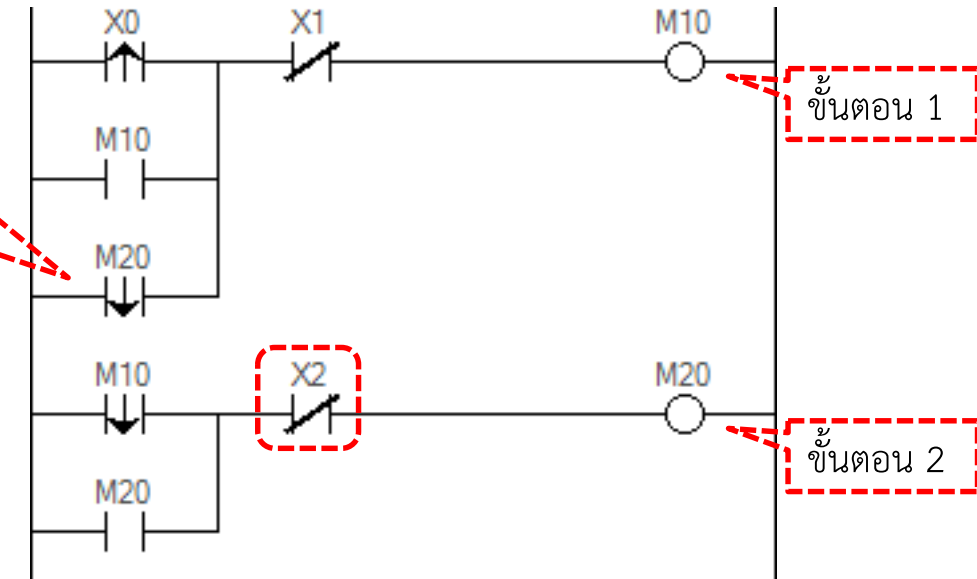
Jump sequence เป็น Ladder ปกติ

กระโดดไปต่อ Step ที่ระบุไว้
Tran2 จริง ทำ Step1



Ladder ปกติ ใช้หน้าสัมผัสตัดวงจร Self hold

จบขั้นตอน 2
กระโดดไปเริ่มขั้นตอน 1



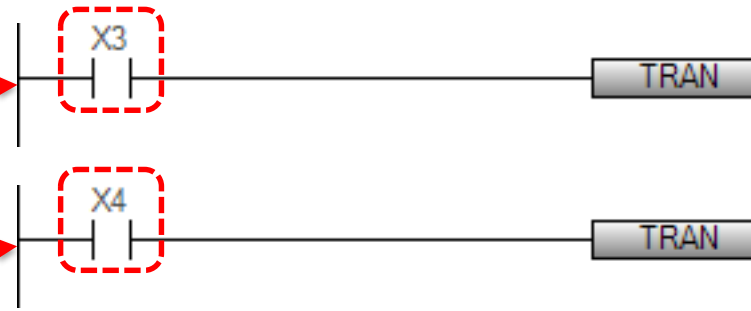
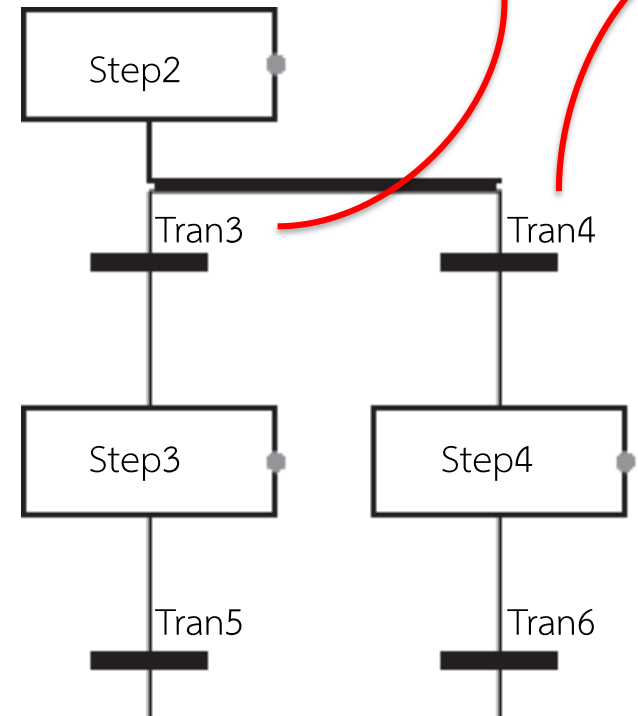
Selective sequence เป็น Ladder ปกติ

เลือกต่อ Step เดียว (ซ้ายสำคัญกว่า)

Tran3 จริง ทำ Step3

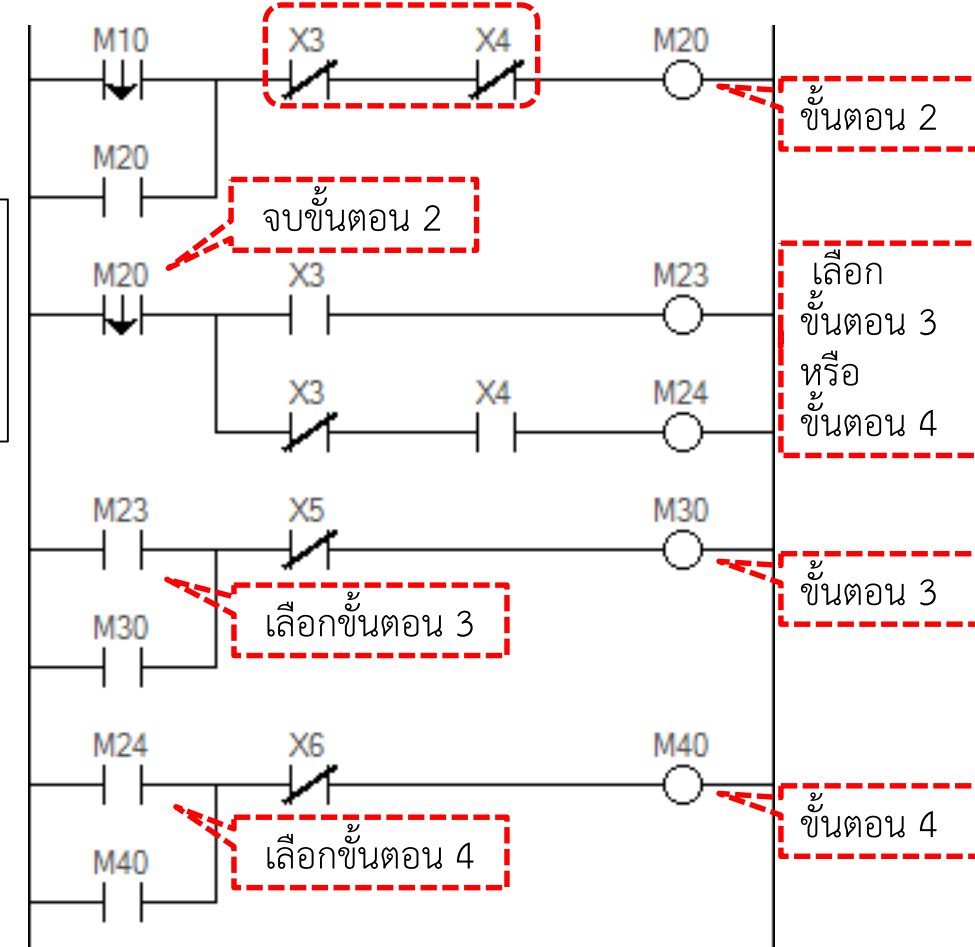
Tran3 ไม่จริง Tran4 จริง

ทำ Step4



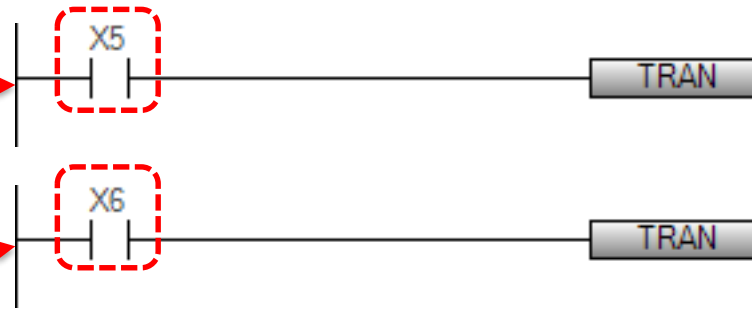
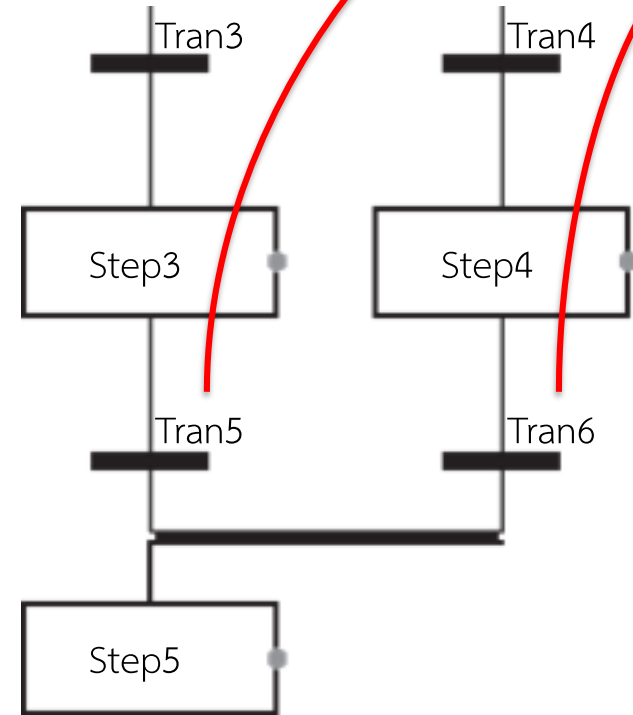
Ladder ปกติ
ใช้การจบขั้นตอนเดิมกับหน้าสัมผัสเงื่อนไข
เลือกขั้นตอนใหม่

Ladder ปกติ ใช้ทุกหน้าสัมผัสตัดวงจร Self hold



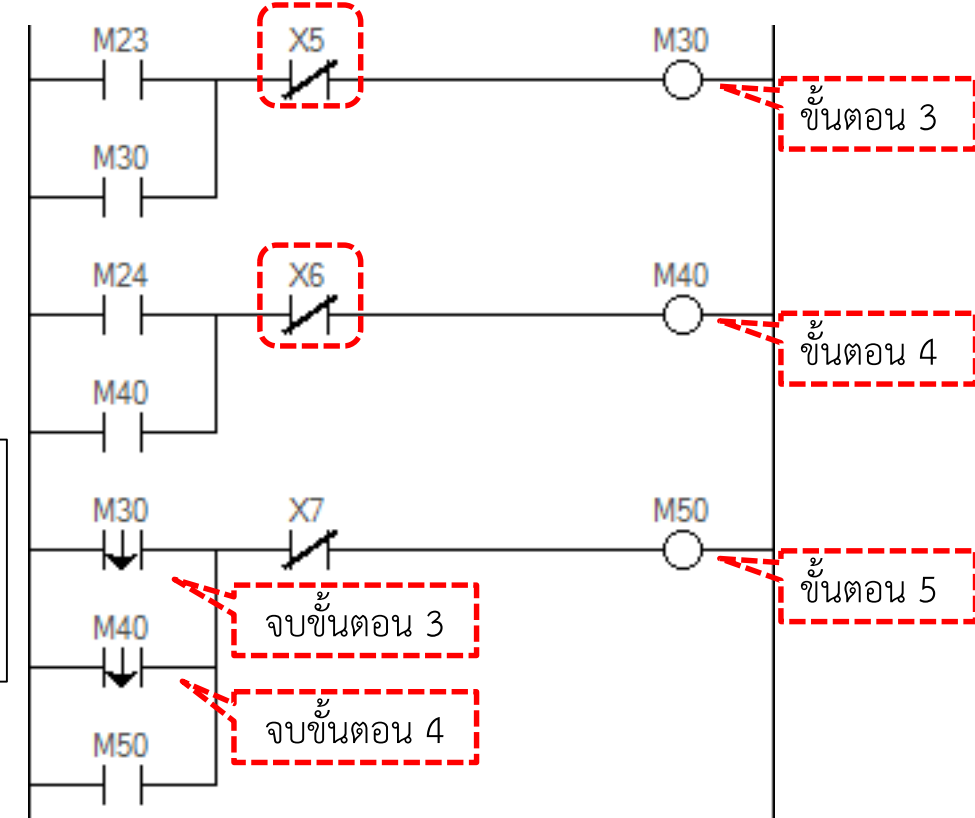
ถ้า Selective sequence มาต่อ Step เดียวกัน

เป็น Series sequence ปกติ
ต่อ Step ที่ตามมา



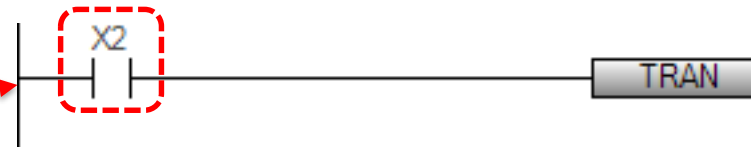
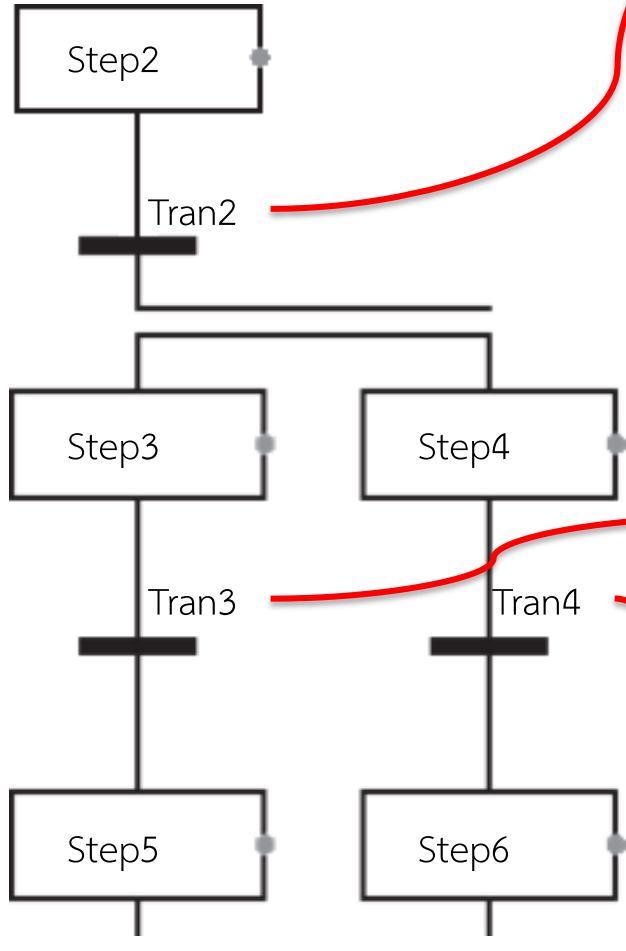
Ladder ปกติ
ใช้การจบขั้นตอนเดิม
สั่งเริ่มขั้นตอนใหม่เดียวกัน

Ladder ปกติ ใช้หน้าสัมผัสตัดแต่ละวงจร Self hold

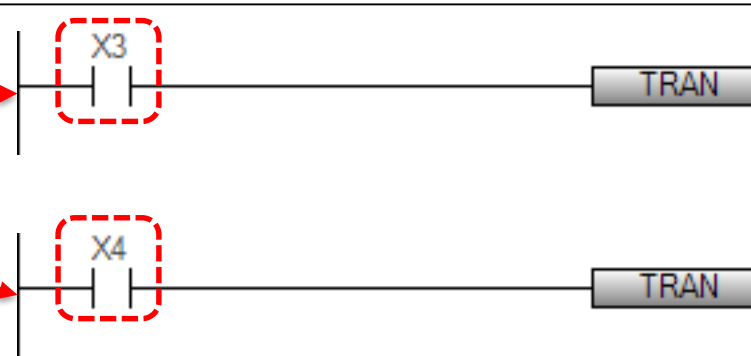


Simultaneous sequence เป็น Ladder ปกติ

ต่อทุก Step ถัดมาพร้อมกัน
Tran2 จริง ทำ Step3 และ Step4

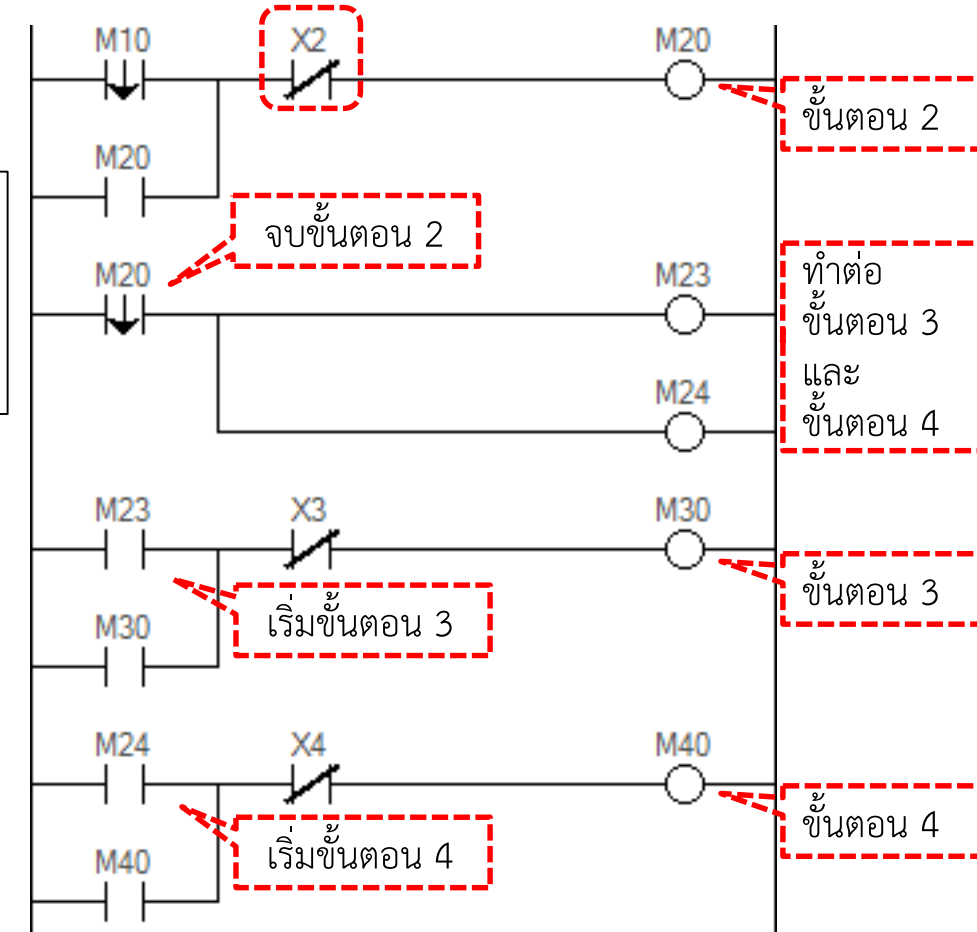


Ladder ปกติ
ใช้การจบขั้นตอนเดิม
เริ่มขั้นตอนถัดไปทุกขั้นตอน



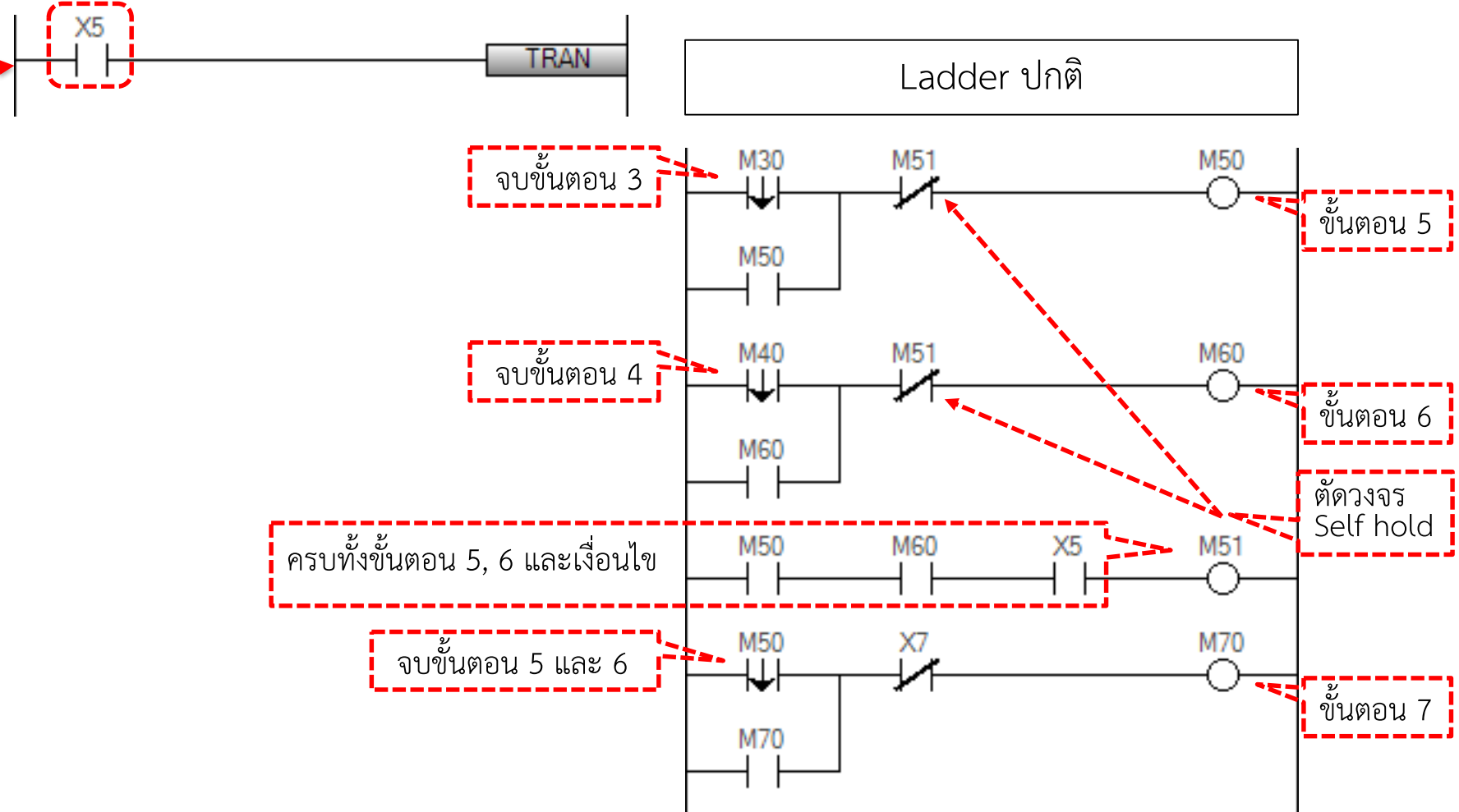
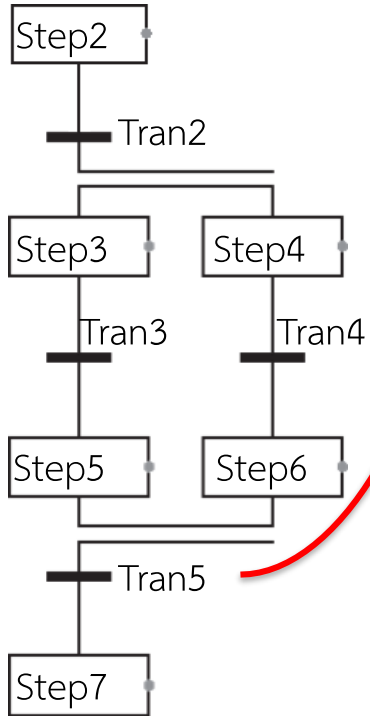
เป็น Series sequence ปกติ
ต่อ Step ที่ตามมา

Ladder ปกติ ใช้หน้าสัมผัสตัดวงจร Self hold



จบ Step สุดท้ายท้ายของ Simultaneous sequence ต้องต่อเป็น Step เดียว

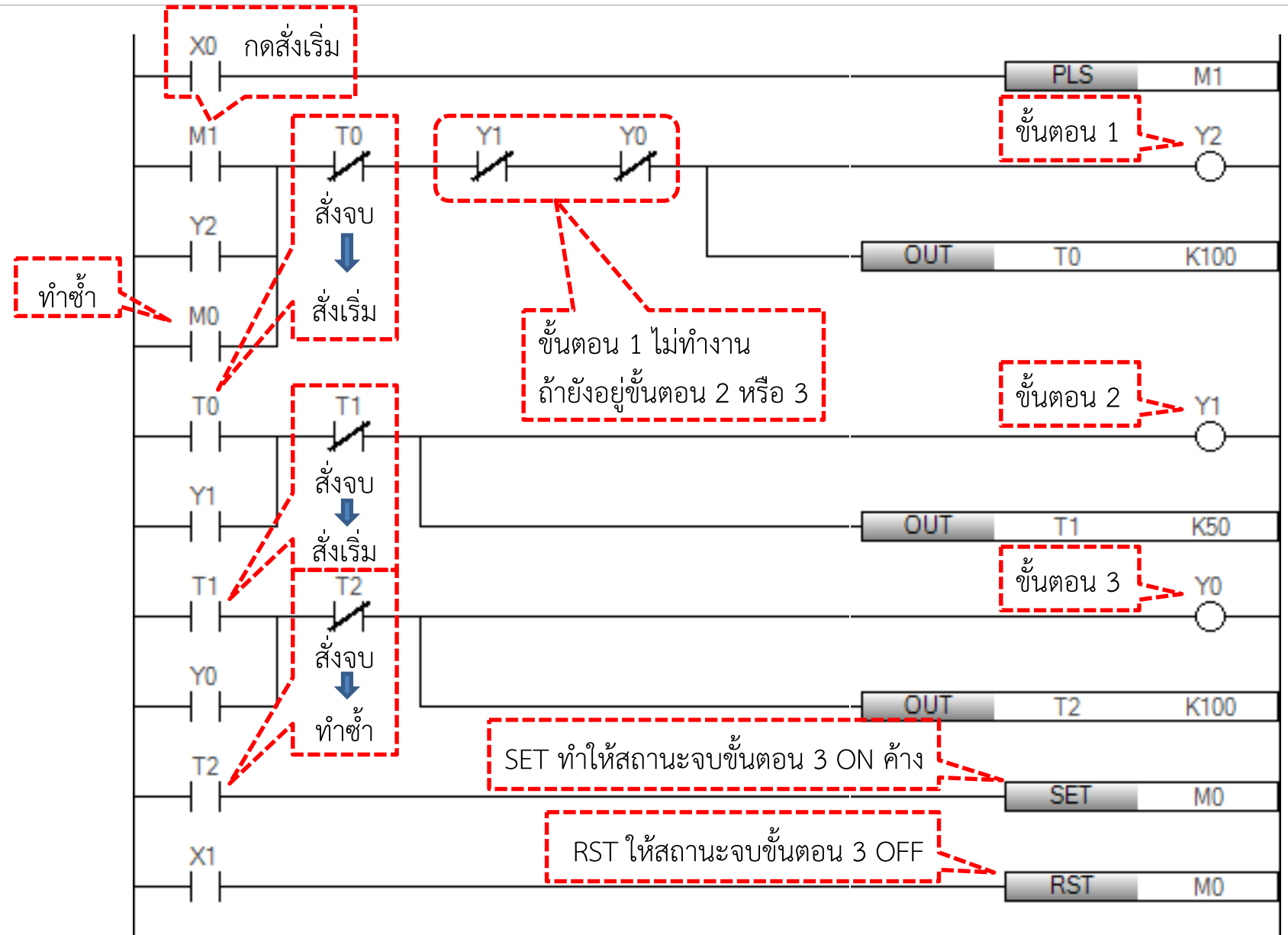
ถ้ากำลังทำทั้ง Step5 และ Step6 แล้ว Tran5 จริง ทำ Step7



ตัวอย่างการโปรแกรม Sequence control

โปรแกรมจากคู่มือบทที่ 5 ข้อ 5.1

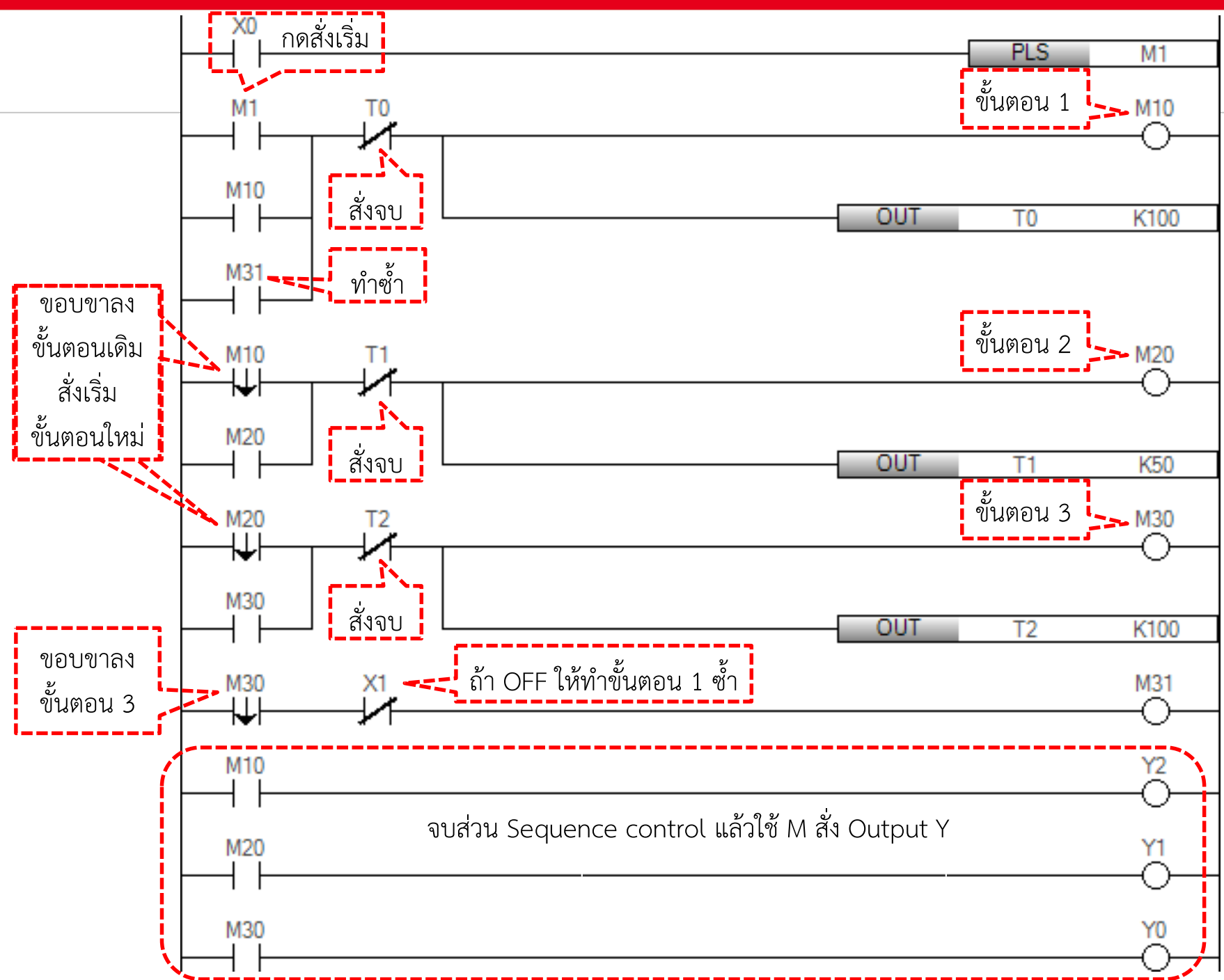
- ใช้ Output Y เป็นขั้นตอนการทำงาน
- Y ค้างสถานะด้วย Self-hold
- สั่งจบขั้นตอนด้วย Timer
- Timer ที่สั่งจบมาสั่งเริ่มขั้นตอนใหม่
- จบขั้นตอน 3 แล้ว SET M0 ให้ทำซ้ำ
- ไม่ทำซ้ำ ให้ ON X1 เพื่อสั่ง RST M0 ในช่วงจบขั้นตอน 3



- ไม่ใช่ Output Y เป็นขั้นตอนการทำงาน
 - เพราะบาง Output อาจ ON ในหลายขั้นตอน
- ใช้ Internal relay M เป็นขั้นตอนแทน
 - ให้หมายเลข M แต่ละขั้นตอนห่างกัน 10 เบอร์ เช่น M10, M20, M30, M110, M120, M130
 - เพื่อเพื่อหมายเลข M สำหรับเงื่อนไขอื่นในแต่ละขั้นตอนให้อยู่ในหมายเลขช่วงเดียวกัน
 - เมื่อจบโปรแกรมส่วน Sequence control แล้วรวมหน้าสัมผัส M ไปสั่ง Output Y
- ไม่ใช่หน้าสัมผัสสั่งจบขั้นตอนเดิมไปสั่งเริ่มขั้นตอนใหม่ ใช้หน้าสัมผัสขอบขาของ M หมายเลขที่เป็นขั้นตอนแทน
 - เพราะหน้าสัมผัสสั่งจบขั้นตอนอาจมีสถานะเดียวกันในขั้นตอนอื่น
 - อาจใช้คำสั่ง PLF ให้ได้ M แทนหน้าสัมผัสขอบขาของขั้นตอนเดิม (หน้าสัมผัสขอบขาจะมีข้อจำกัดในการใช้ซ้ำ)
- ส่วน Sequence ใช้วงจร Self hold เป็นหลัก
- ไม่ใช่คำสั่ง SET/RST โดยไม่จำเป็น
 - มีโอกาสทำให้ ON นานเกินจำเป็นและต้องแก้ปัญหาด้วยวิธีอื่นมาช่วยอีกตามโปรแกรมตัวอย่างข้อ 5.1
 - การใช้ SET/RST ให้ ON หรือ OFF ได้หลายทีในโปรแกรม จะหาที่มาของปัญหายาก
 - ยังต้องใช้ RST กับ Retentive timer หรือ Counter

โปรแกรมข้อ 5.1 ที่ปรับแก้แล้ว

- ใช้ M10, M20, M30 เป็นขั้นตอน
- สั่งจบขั้นตอนด้วย Timer
- ใช้ของขาลงของขั้นตอนเดิม
สั่งเริ่มขั้นตอนใหม่
- จบขั้นตอน 3
 - ถ้า X1 OFF จะให้ M31 ON เพื่อทำซ้ำ
 - ถ้า X1 ON จะไม่ทำซ้ำ
- จบส่วน Sequence control
 - ใช้ M สั่ง Output Y

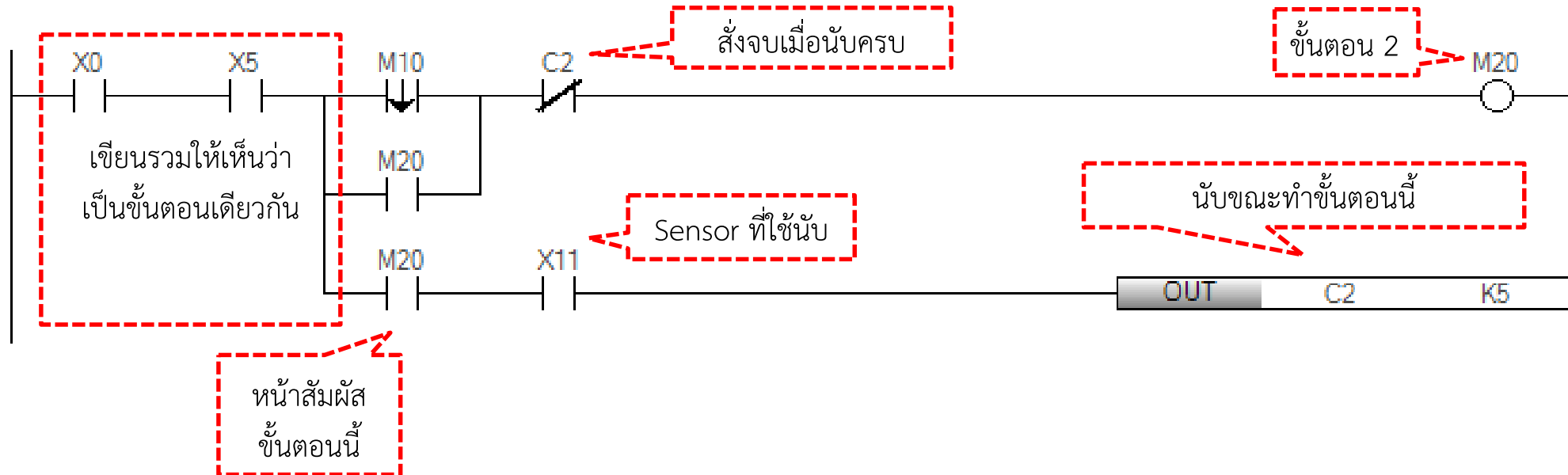


การโปรแกรม Sequence control ให้เข้าใจลำดับง่าย

- เขียนหน้าสัมผัสที่ใช้ร่วมกับขั้นตอนอื่นก่อนวงจร Self hold ในแต่ละขั้นตอน เช่น Manual/Auto mode, Emergency stop



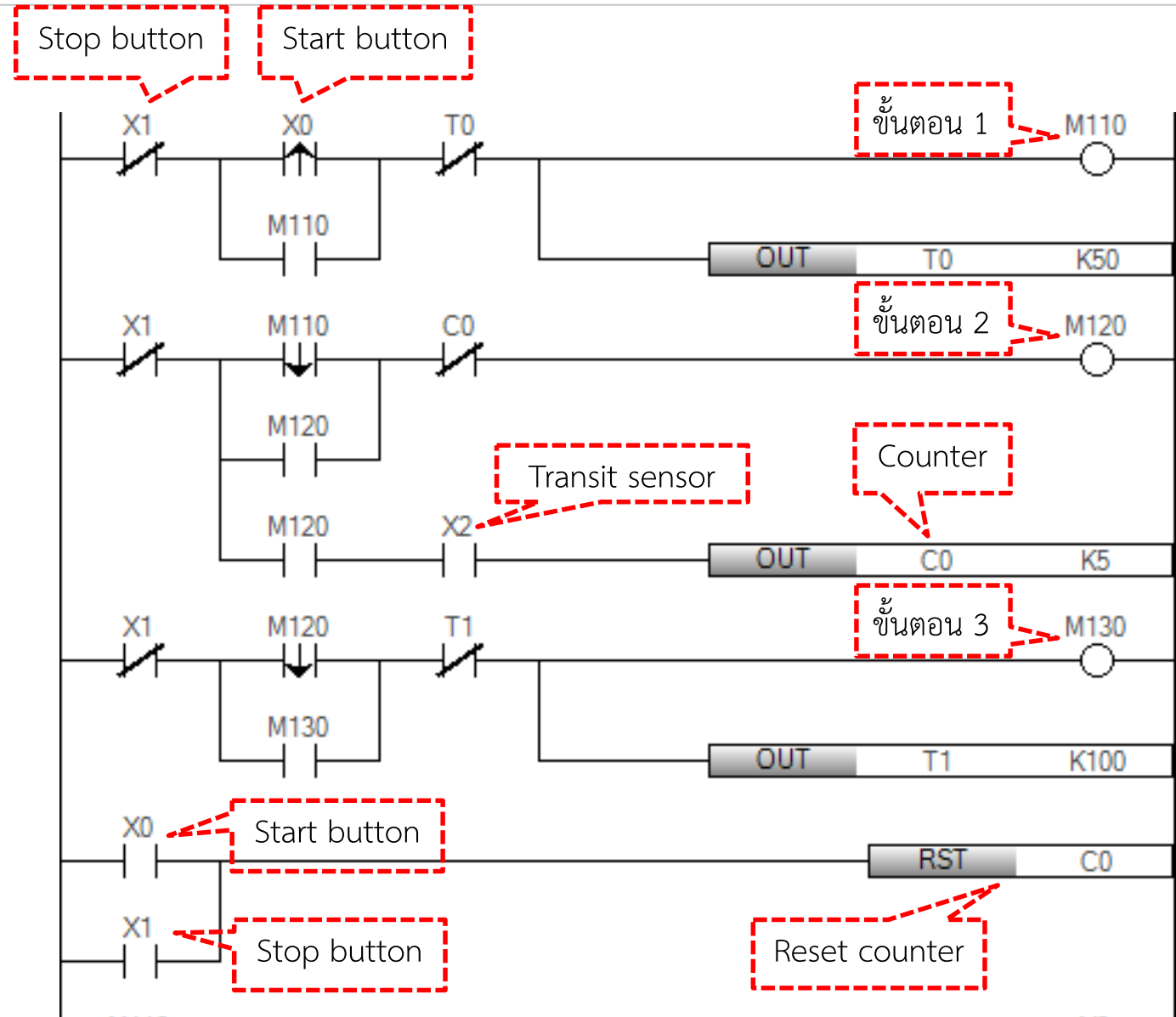
- เขียนโปรแกรมที่ต้องใช้เฉพาะขั้นตอนนั้นในหน้าสัมผัสร่วมกับวงจร Self hold เช่น การนับจำนวนขณะทำขั้นตอน



โปรแกรมข้อ 5.2 ที่ปรับแก้แล้ว (1)

คิดโปรแกรมใหม่เป็น 3 ขั้นตอน

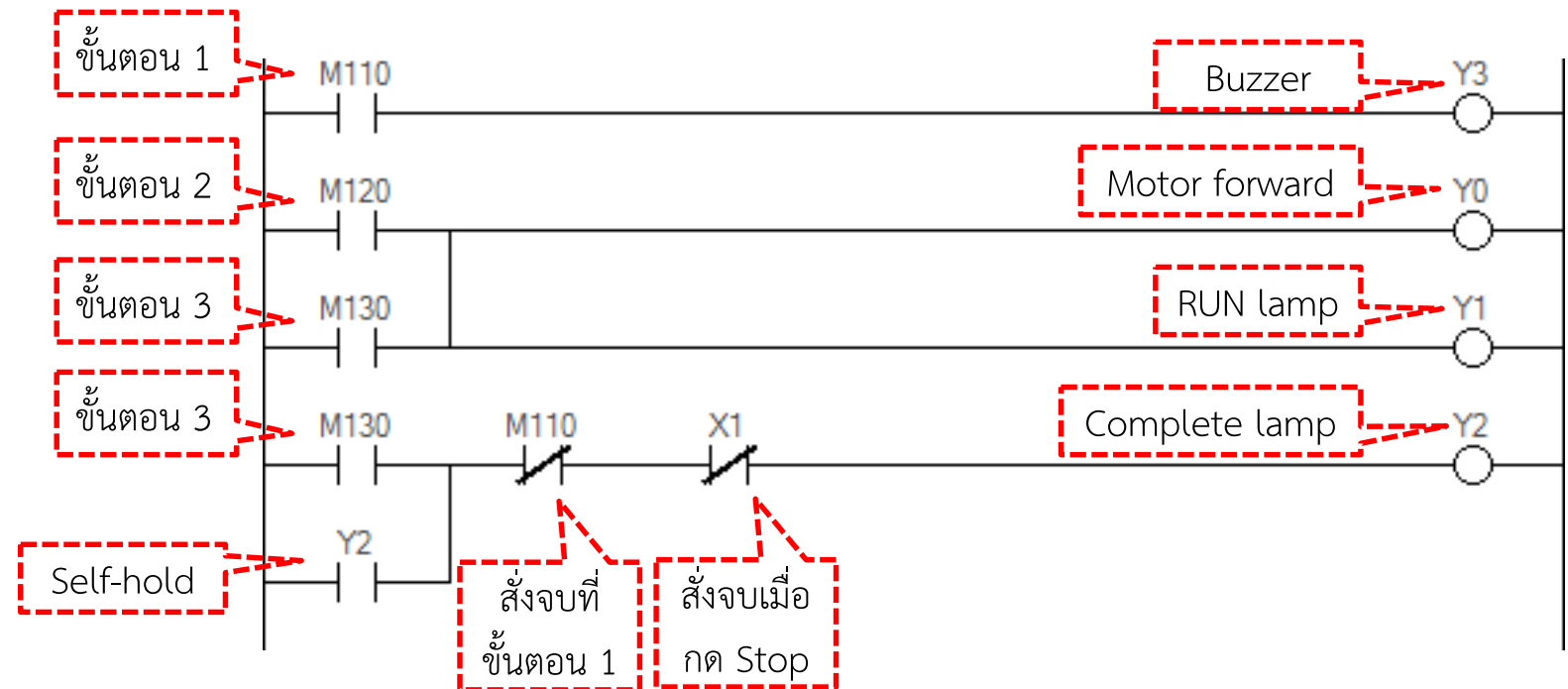
- ใช้ Stop button ร่วมกันในทุกขั้นตอน
- กด Start button เริ่มขั้นตอนแรก
- ขั้นตอน 1; M110
 - Buzzer ดังจนจบเวลา
- ขั้นตอน 2; M120
 - Motor forward ทำงาน
 - จนนับ Transit sensor ครบ
- ขั้นตอน 3; M130
 - Complete lamp เริ่มติด
 - Motor forward ยังทำจนจบเวลา
- Reset Counter เมื่อกดปุ่ม Start หรือ Stop

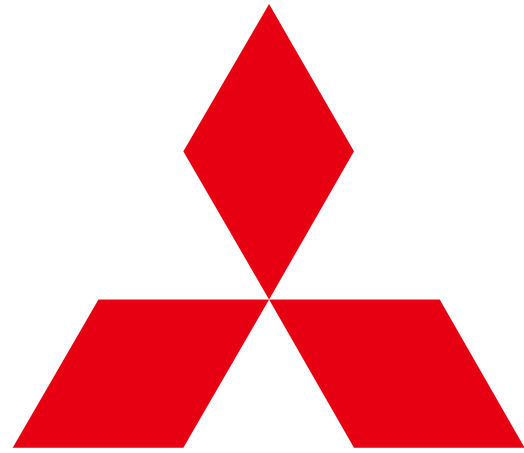


โปรแกรมข้อ 5.2 ที่ปรับแก้แล้ว (2)

ส่วนให้ Output ทำงาน

- Buzzer
 - ทำงานในขั้นตอน 1
- Motor forward และ RUN lamp
 - ทำงานในขั้นตอน 2
 - และขั้นตอน 3
- Complete lamp
 - เริ่มทำงานที่ขั้นตอน 3
 - จบเมื่อทำขั้นตอน 1 หรือกดปุ่ม Stop





**MITSUBISHI
ELECTRIC**

Changes for the Better