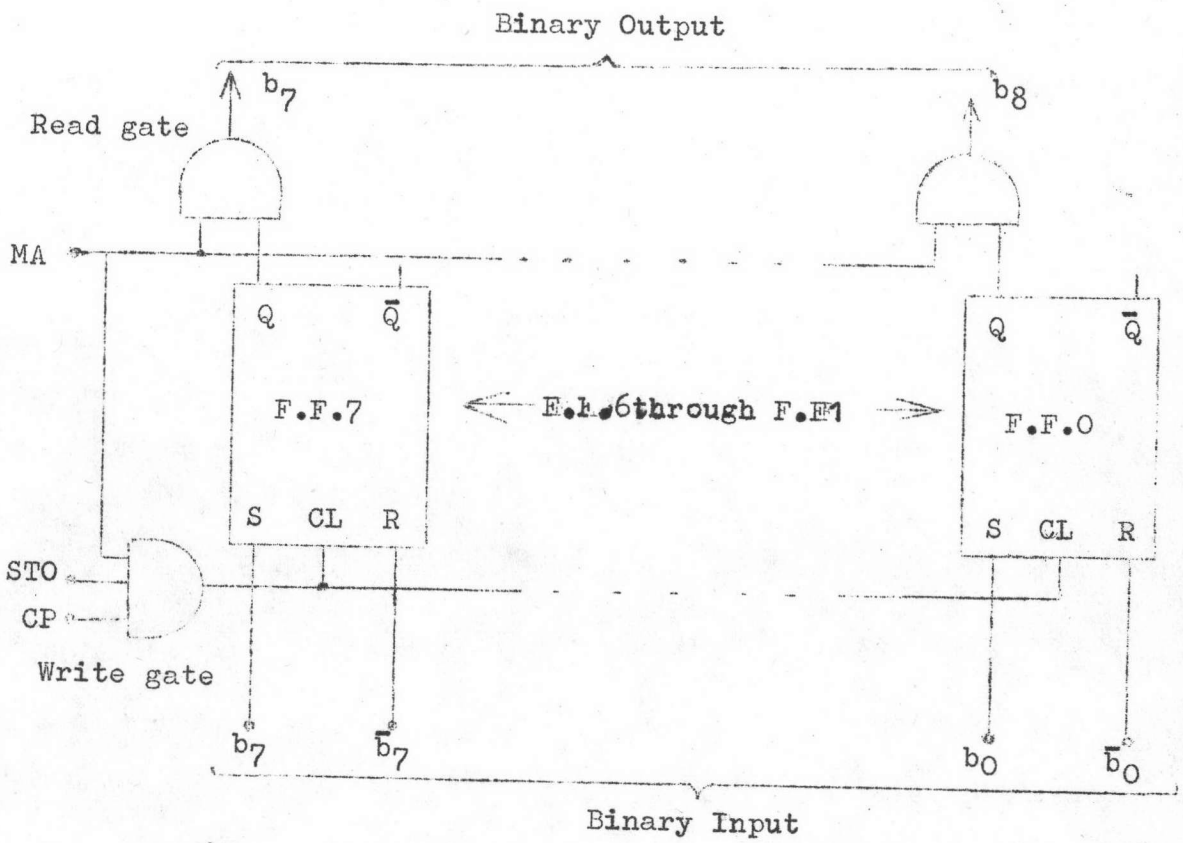
บทที่ ๓

การสร้างโลจิคฟังค์ชั่นและไดอะแกรม

๓.๑ หน่วยความจำ

๓.๑.๑ Memory Register and Read/Write Control Gate

ใน Memory register 1 word ประกอบด้วย Clocked-RS flip flop ๘ ตัว clocked-input ของ flip flop จะต่อร่วมกันและถูกควบคุมด้วย write control gate ดังรูปที่ ๓.๑ RS-input แต่ละ bit จะต่อมาจากภาค input driver ข้อมูลจะ transfer เข้าไปใน register ได้เมื่อ address (MA), คำสั่ง Store (STO)และ clock pulse เป็น logic "๑"

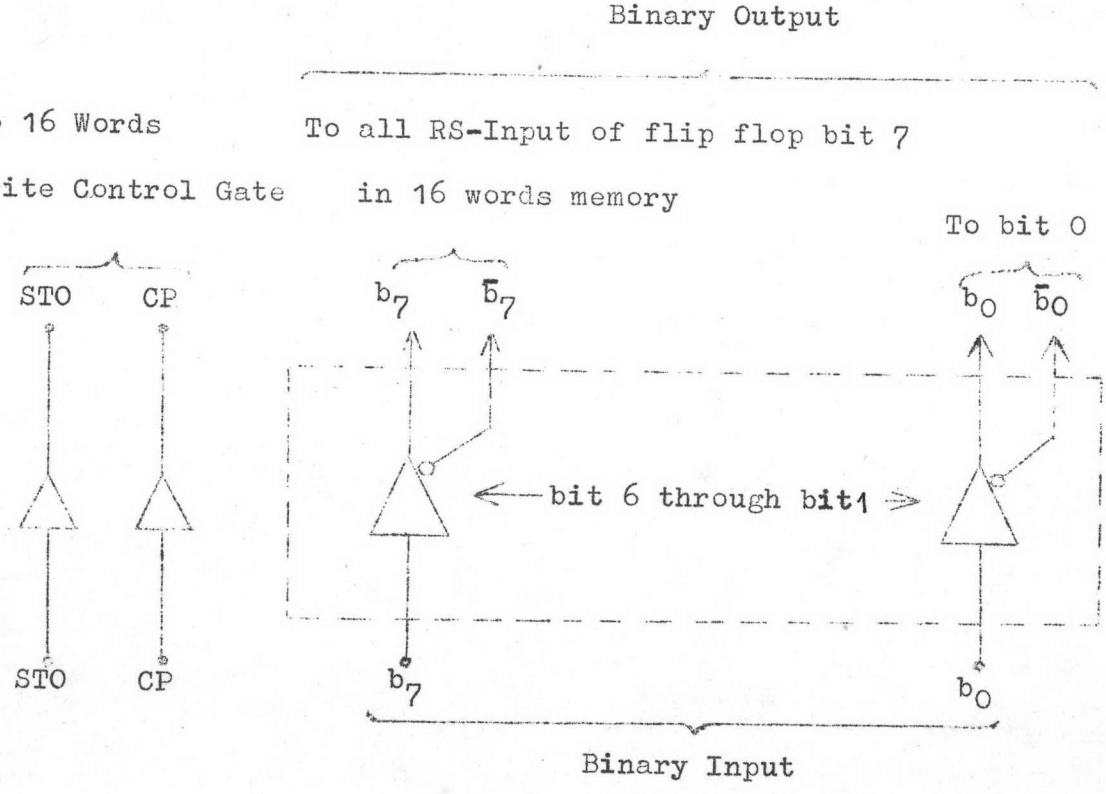


รูปที่ ๓.๑ Logic Diagram of Memory Register

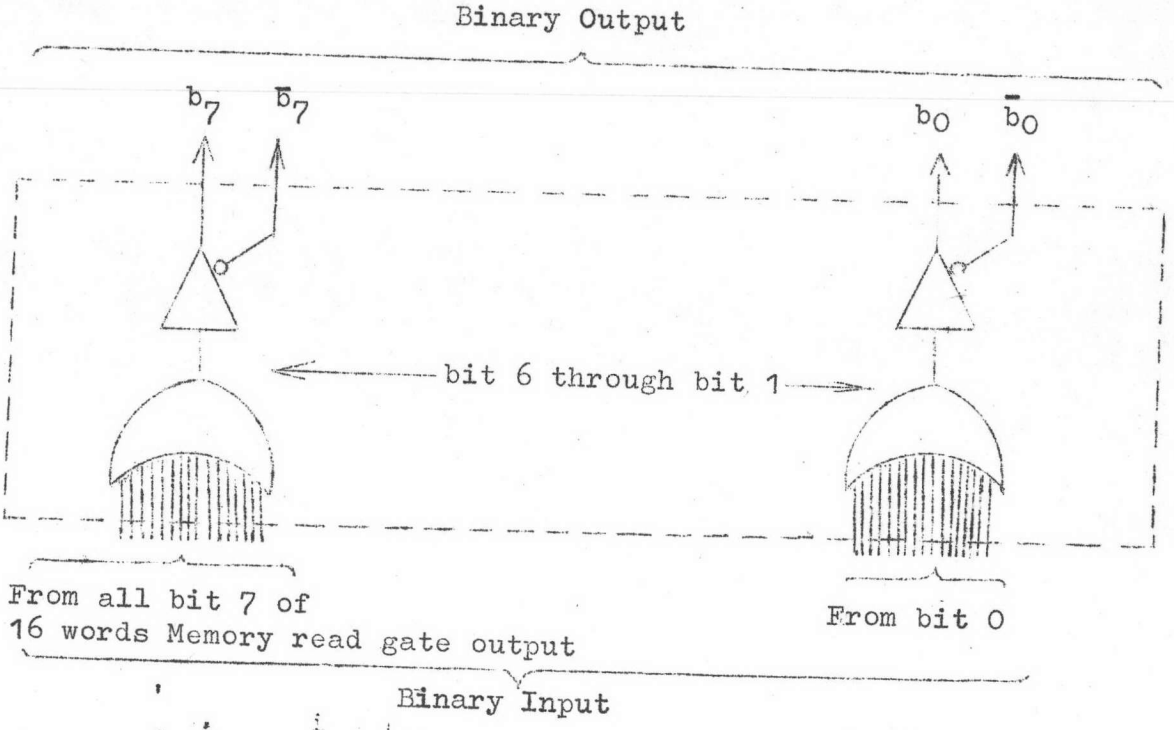
สำหรับการอ่านข้อมูลออกจาก register นั้น จะถูกควบคุมด้วย Read gate ซึ่งข้อมูลใน register จะออกได้เมื่อ logic ของ MA เป็น "1" เท่านั้น

๓.๑.๒ Memory Input/Output Driver

เนื่องจากหน่วยความจำมีทั้งหมด ๑๖ คำ ดังนั้นทาง input และ output จึงจำเป็นจะต้องมีภาค driver ทั้งนี้เพื่อป้องกัน loading effect Input driver จะมีลักษณะดังรูปที่ ๓.๒ ซึ่ง input ของวงจรจะรับข้อมูลมาจาก Accumulator และทาง output จะต่อเข้าไปยัง input ของ memory register แต่ละชุด สำหรับ Output driver จะมีลักษณะดังรูปที่ ๓.๓ ซึ่งจะรับข้อมูลในแต่ละ bit ของ register ทั้ง ๑๖ คำ แล้ว drive ไปยัง Buffer register และ Program register



รูปที่ ๓.๒ Logic Diagram of Input Memory Driver



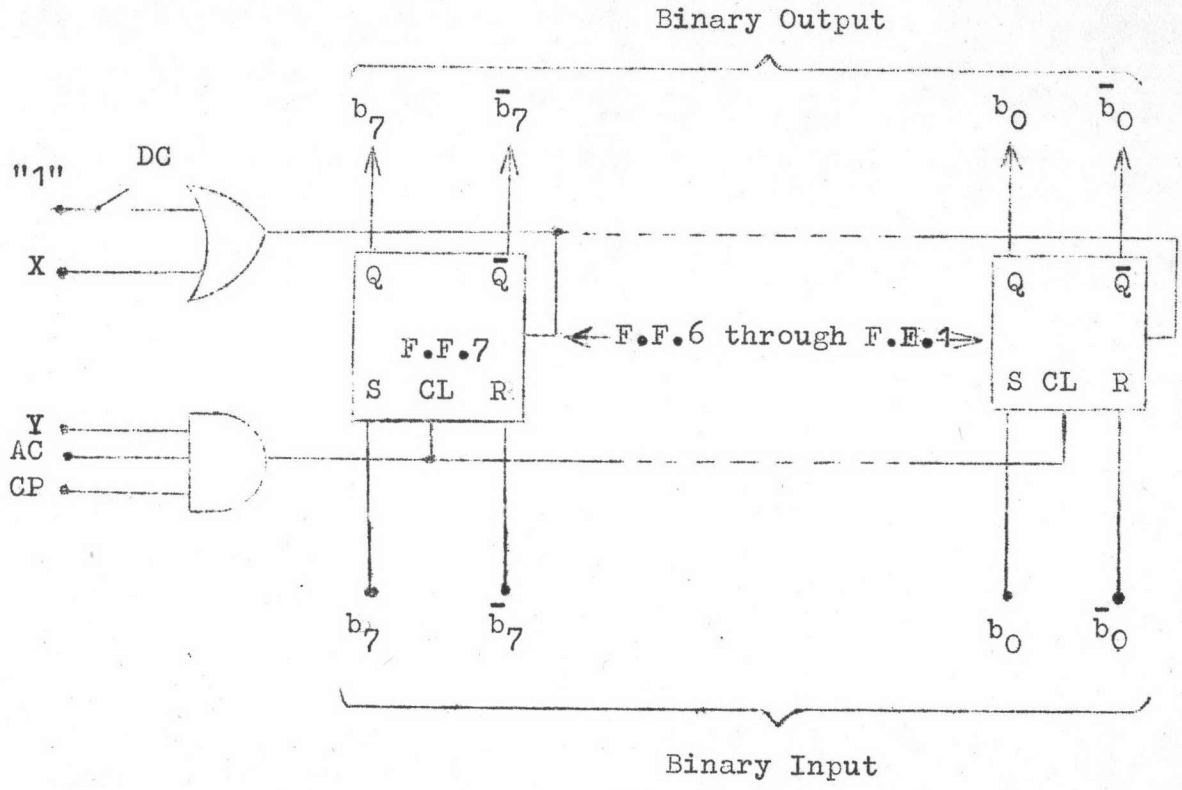
รูปที่ ๓.๓ Logic Diagram of Output Memory Driver

๓.๒ หน่วยคำนวณ

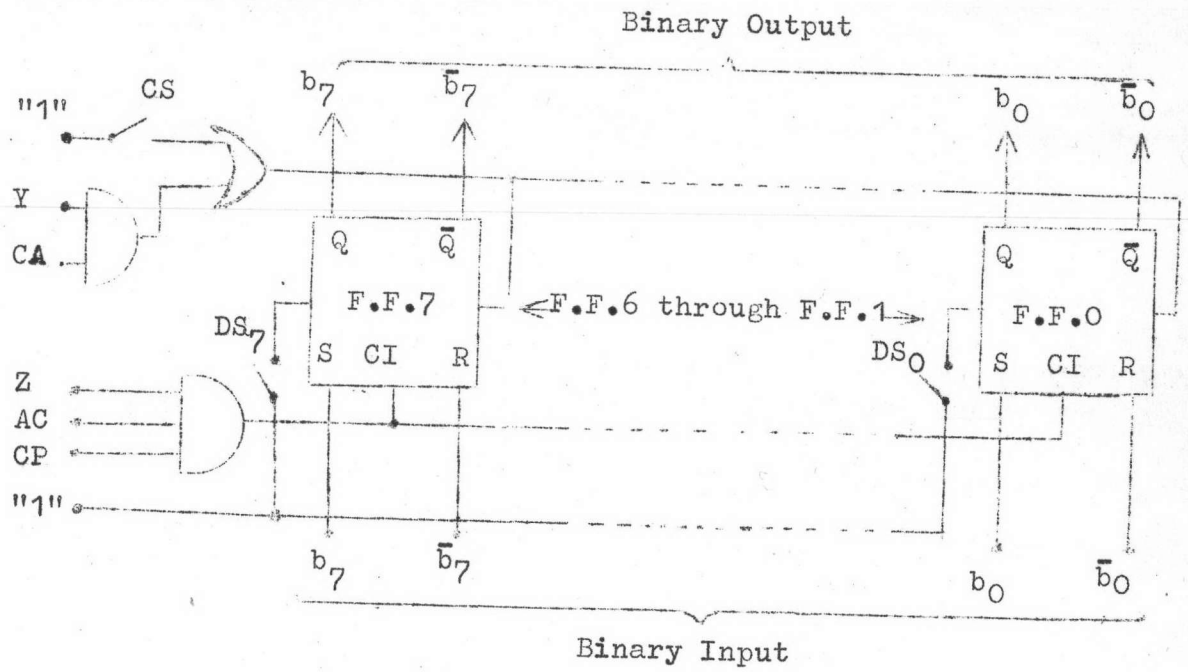
๓.๒.๑ Buffer Register and Accumulator

Buffer register และ Accumulatorจะประกอบด้วย clocked-RS flip flop ๘ ตัว clocked-input ของ flip flop จะต่อร่วมกันและถูกควบคุมด้วย input control gate ดังรูปที่ ๓.๔ และ ๓.๕ ตามลำดับ สำหรับ Buffer register ข้อมูลจากหน่วยคำนวณจะ transfer เข้าไปในregister ได้ก็ต่อเมื่อการทำงานของเครื่องอยู่ใน phase Y และมีคำสั่งเกี่ยวกับการคำนวณ (AC) เท่านั้น ส่วน Accumulator ก็เช่นเดียวกันกับ Buffer register แต่การ transfer จะเกิดขึ้นเมื่อการทำงานของเครื่องอยู่ใน phase Z (ดังปรากฏอยู่ในตารางที่ ๒.๑) ใน Buffer register จะมีวงจรสำหรับ clear ทั้งทาง logic และ switch ซึ่งจากตารางที่ ๒.๑ ในช่วงของ phase X ข้อมูลจะถูก clear เสมอ ดังนั้นจึงควบคุม clear ด้วย

phase X pulse ถ้าต้องการ clear ในขณะที่เครื่องทำงานอยู่ใน phase อื่น ก็สามารถทำได้โดยใช้ clear switch สำหรับใน Accumulator ก็จะมีวงจร clear เช่นเดียวกัน แต่การ clear จะเกิดขึ้นในขณะเครื่องอยู่ใน phase Y และมีคำสั่ง Clear and Add หรือ Clear and Subtract เท่านั้น นอกจากนี้ยังมี Clear Switch และ Set Switch สำหรับ unload และ load ข้อมูลหรือโปรแกรมอีกด้วย



รูปที่ ๓.๔ Logic Diagram of Buffer Register



รูปที่ ๓.๕ Logic Diagram of Accumulator

๓.๒.๒ One's Complement Transfer Gate

ในกรณีที่มีคำสั่ง Subtract หรือ Clear and Subtract Output ของ Buffer register จะถูกเปลี่ยนให้อยู่ในรูปของ One's complement ก่อนที่จะบวกกับข้อมูลใน Accumulator โดยใช้วงจร 8 bit One's complement transfer gate แต่ละ bit จะมีฟังค์ชั่นการทำงานดัง truth table ในตารางที่ ๓.๑ ซึ่งนำไปสร้างวงจรได้ดังรูปที่ ๓.๖

B	T	F
0	0	0
0	1	1
1	0	1
1	1	0

Notes : B = Binary input form BR

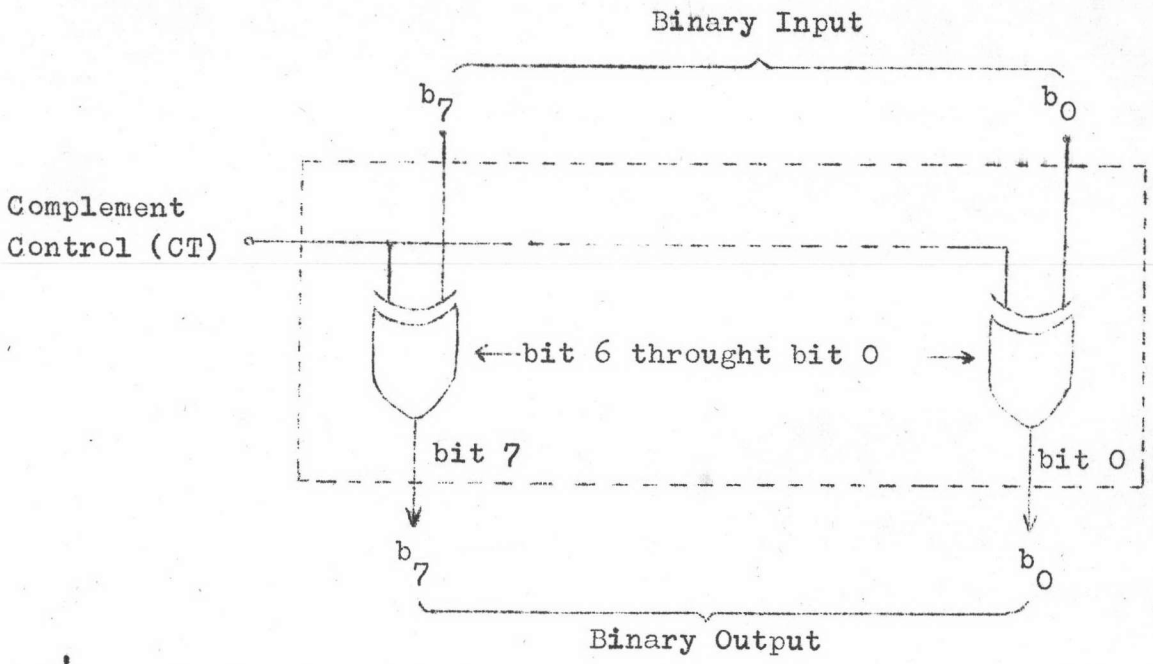
T = Subtract control

F = Binary output

Function: $F = \bar{B}T + \bar{T}B$

$= B \oplus T$

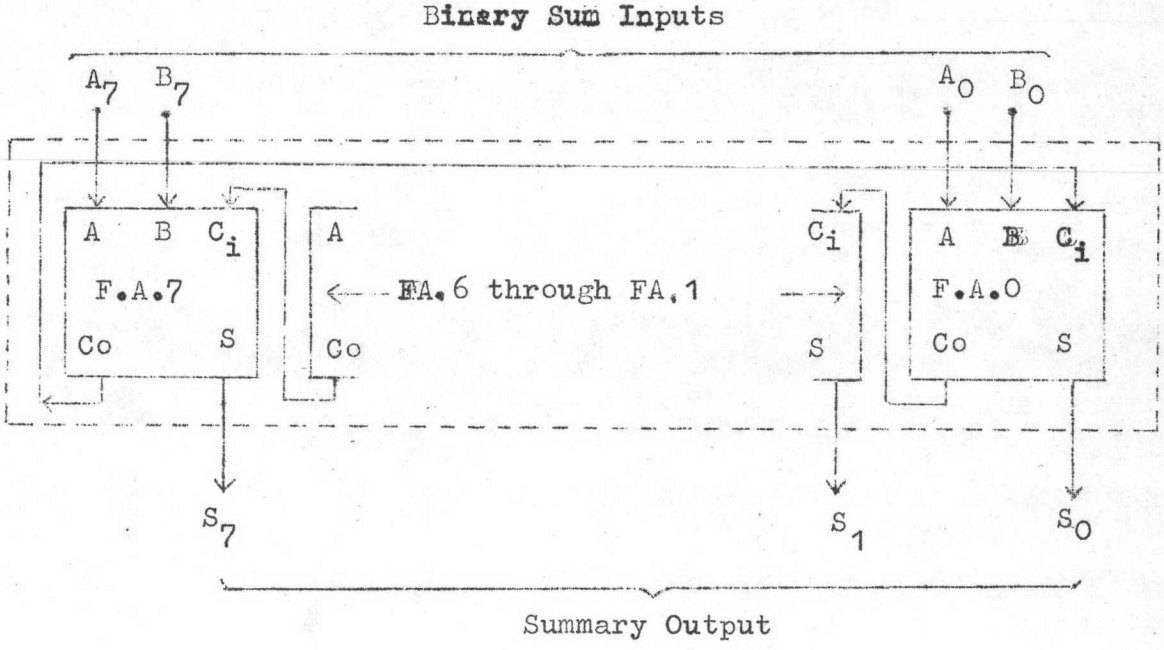
ตารางที่ ๓.๑ Truth Table for each bit of One's Complement Gate



รูปที่ ๓.๖ Logic Diagram of One's Complement Transfer Gate

๓.๒.๓ Parallel Full Adder

Parallel Full Adderประกอบด้วยวงจร full adder 8 bit ต่อ cascade กัน ดังรูปที่ ๓.๗ โดยที่ carry outputของ bit 0 ต่อเข้ากับ carry input ของ bit 1 และของ bit 1 ต่อไปยังbit 2 ตามลำดับ สำหรับ carry output ของ bit สุดท้าย คือ bit 7 จะต่อกลับไปยัง carry input ของ bit 0 ทั้งนี้เพื่อให้การลบโดยวิธีบวก one's complement จบสมบูรณ์ในตัวมันเอง (พิจารณาภาคผนวก ก ประกอบ) ฟังค์ชั่นการทำงานของ full adder แต่ละ bit จะมีลักษณะดังตารางที่ ๓.๒ การสร้างวงจร full adder จะใช้ half adder เป็นตัวสร้างวงจรดังรูปที่ ๓.๘



รูปที่ ๓.๗ Logic Diagram of Parallel Full Adder

A	B	C_i	S	C_o
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

Note : A = Binary input from Accumulator

B = Binary input from Buffer register

C_i = Carry input

S = Sum output

C_o = Carry output

Function :

$$S = A \oplus B \oplus C_i$$

$$C_o = AB + C_i(A \oplus B)$$

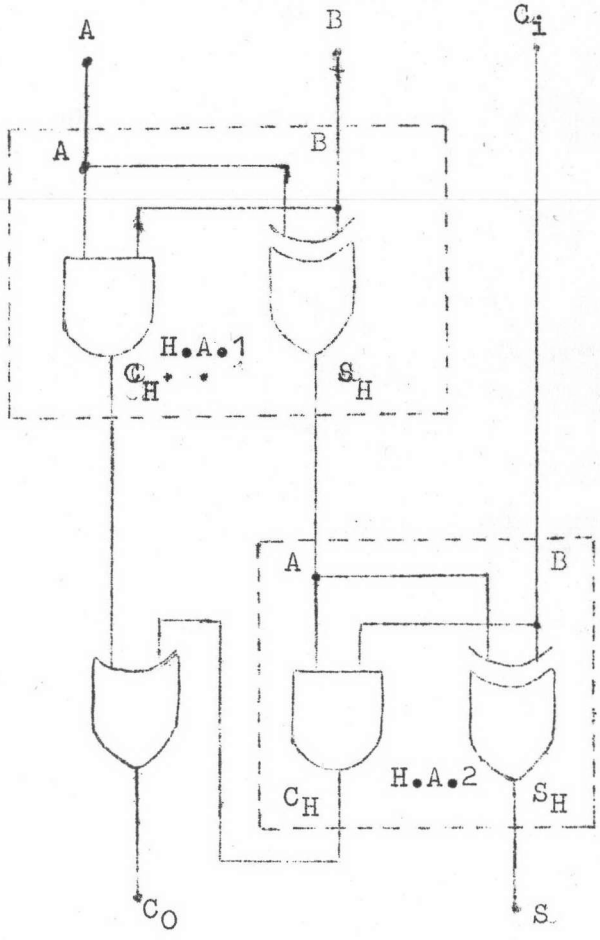
ตารางที่ ๓.๒ Truth Table for each bit of Paralle Full Adder

A	B	S_H	C_H
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

$S_H = A \oplus B$

$C_H = AB$

ก) Truth Table of Half Adder



ข) Full Adder Implementation

รูปที่ ๓.๘ Logic Diagram of Full Adder

๓.๒.๔ Overflow Checker

การตรวจ overflow ทำได้โดยเปรียบเทียบเครื่องหมายของตัวตั้งใน Accumulator (S_a) เครื่องหมายของตัวบวกหรือลบใน Buffer register หลังจาก complement แล้ว (S_c) และเครื่องหมายของผลลัพธ์ที่ได้จาก parallel full adder (S_f) ว่าได้ผลดังในตารางที่ ๓.๓ หรือไม่ ถ้าหากว่าผิดไปจากตารางนี้แล้วแสดงว่าเกิด overflow ในทาง logic ถ้า Sign bit เป็น 1 หมายถึงเครื่องหมายลบ ถ้า Sign bit เป็น 0 หมายถึงเครื่องหมายบวก ฟังค์ชั่นการทำงานของวงจรจะเขียนได้ดัง Truth table ตารางที่ ๓.๔ โดยถือว่า ถ้าเครื่องหมายเหมือนกันบวกกันแล้วผลลัพธ์

ได้เครื่องหมายตรงข้ามแสดงว่าผลของการบวกเกินหลัก ถ้าเครื่องหมายต่างกันบวกกัน โอกาสที่เกินหลักจะไม่มีเลย (พิจารณาภาคผนวก ก ประกอบ) การตรวจสอบ over-flow จะมีในระหว่างที่เครื่องทำงานอยู่ใน phase Y และได้รับคำสั่งเกี่ยวกับการบวกหรือลบเท่านั้นตามตารางที่ ๒.๑ ข้อ b) และ c) Output ของวงจรนี้จะไป set overflow flip flop เพื่อแสดงผลและทำให้เครื่องหยุดทำงานดังแสดงในรูปที่ ๓.๙

The larger number	The smaller number	The results of sum
+ X	+ Y	+(X+Y), the result is greater than X
+ X	- Y	+(X-Y), the result is smaller than X
- X	+ Y	-(X-Y), the result is smaller than X
- X	- Y	-(X+Y), the result is greater than X

ตารางที่ ๓.๓ Algorithm of Addition for Number with Sign

S_a	S_c	S_f	OF
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	0
1	0	1	0
1	1	0	1
1	1	1	0

Note: 0 = Positive sign

1 = Negative sign

S_a = The sign of addend

S_c = The sign of augend

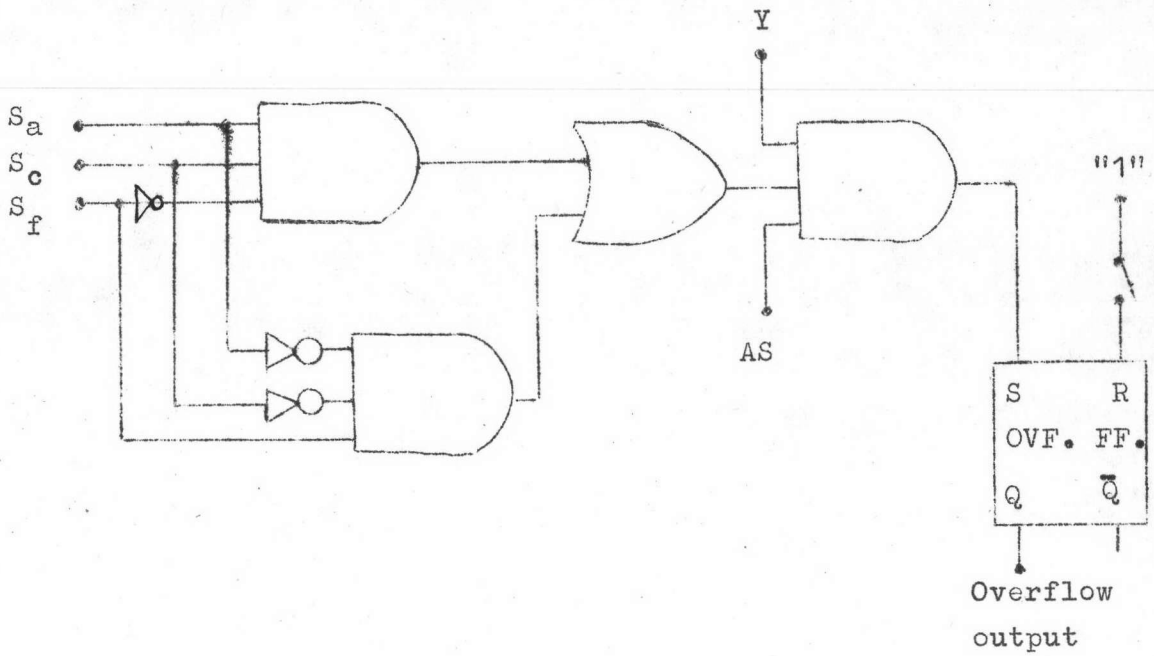
S_f = The sign of result

OF = Overflow function

The number is operated in the form of signed-1's complement representation

Function: $OF = \bar{S}_a \bar{S}_c S_f + S_a S_c \bar{S}_f$

ตารางที่ ๓.๔ Truth Table of Overflow Checker

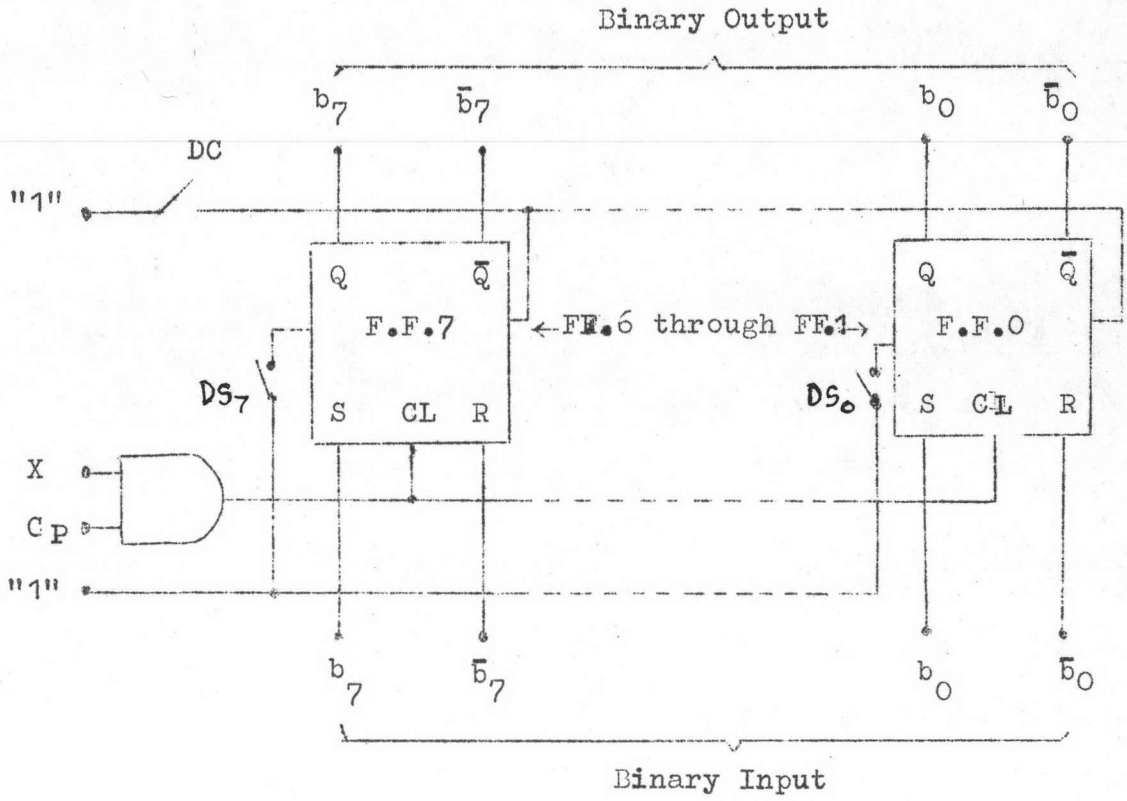


รูปที่ ๓.๙ Logic Diagram of Overflow Checker

๓.๓ หน่วยควบคุมคำสั่ง

๓.๓.๑ Program Register

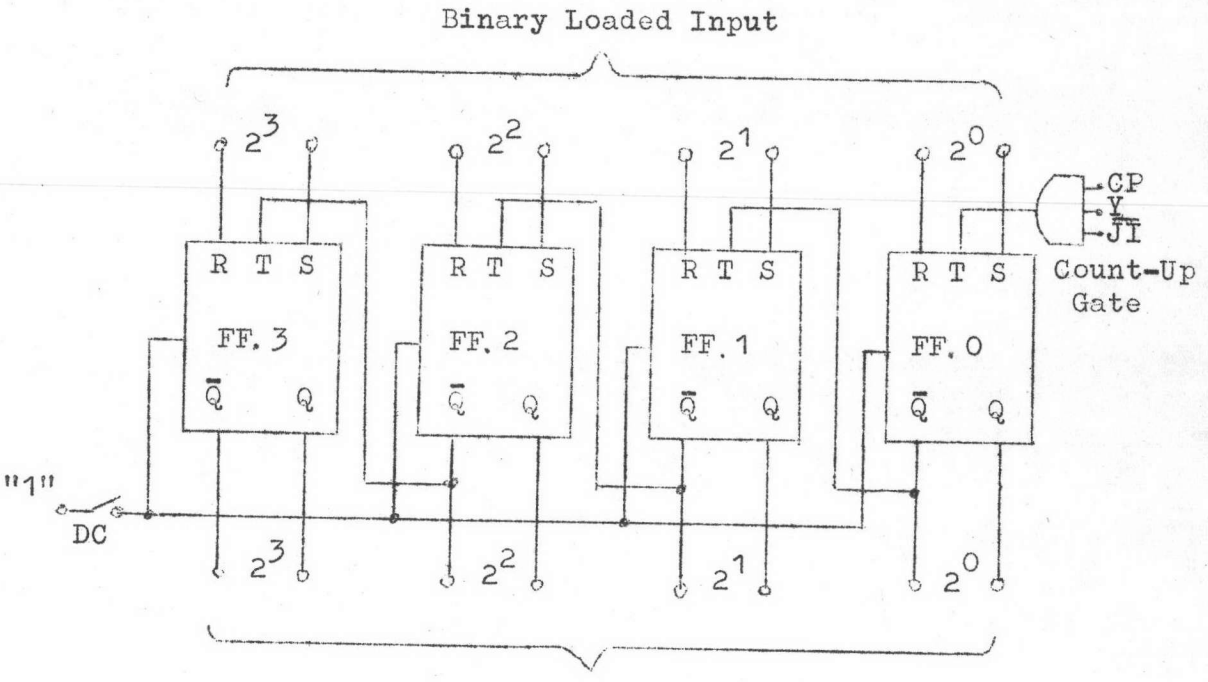
Program registerประกอบด้วย clocked-RS flip flop ๘ ตัวเช่นเดียวกับ Buffer register หรือ Accumulator แต่การ transfer data เข้า จะเกิดขึ้นในช่วงที่เครื่องทำงานใน phase X (Program time) ซึ่งจะมีไดอะแกรมดังรูปที่ ๓.๑๐ ในวงจรนี้สามารถ unload หรือ load program ได้ด้วย clear และ set switches เช่นเดียวกับ Accumulator ทั้งนี้เพื่อใช้สำหรับตั้งคำสั่งต่าง ๆ เพื่อตรวจสอบระบบและใช้เขียน program เข้าไปเก็บไว้ใน Memory register



รูปที่ ๓.๑๐ Logic Diagram of Program Register

๓.๓.๒ Address Register

Address register ประกอบด้วย Triggering flip flop ๔ ตัว ทำเป็นวงจร 4 bit binary count-up counter ดังรูปที่ ๓.๑๑ การนับขึ้นทุก ๆ ครั้งจะอยู่ใน phase Y และไม่มี Jump condition เกิดขึ้นเท่านั้น ถ้ามี Jump condition เกิดขึ้น Address code ใน program register จะผ่าน transfer gate 2มา load วงจรให้ได้ code ตามที่ต้องการ ในวงจรนี้จะมี clear switch สำหรับ set output ให้เป็น 0000 เพื่อเริ่มต้นการทำงานของเครื่อง



รูปที่ ๓.๑๑ Logic Diagram of Address Register

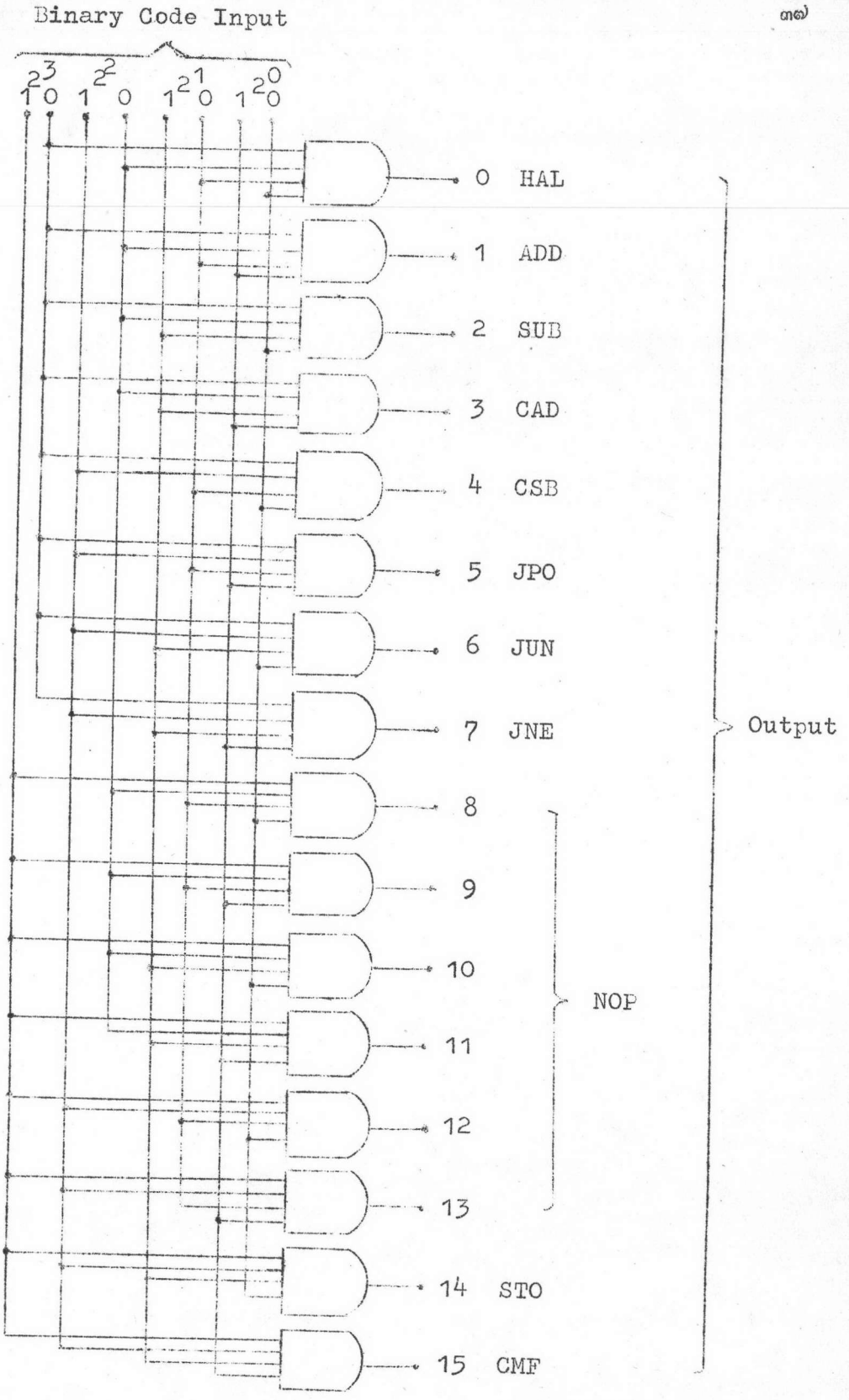
๓.๓.๓ Address and Function Decoder

Address and Function Decoder ประกอบด้วยวงจร 4-to-16-line decoder โดยแปลง address code และ operation code จาก 4 bit binary ให้เป็น address 16 numbers และคำสั่ง 16 instructions ตามลำดับ output ของ Address decoder แต่ละ line จะต่อไปยัง write gate ของ Memory register แต่ละ word และ output ของ function decoder แต่ละ line จะถูก form เป็น instruction control gate ซึ่งใช้ควบคุมการทำงานของเครื่อง ฟังค์ชั่นการทำงานของ decoder จะมีลักษณะดังตารางที่ ๓.๕ และรูปที่ ๓.๑๒

I16804762

Binary Code				Output Function	Description	
b_3	b_2	b_1	b_0		Address No.	Instruction
0	0	0	0	$\bar{b}_3\bar{b}_2\bar{b}_1\bar{b}_0$	0	HAL
0	0	0	1	$\bar{b}_3\bar{b}_2\bar{b}_1 b_0$	1	ADD
0	0	1	0	$\bar{b}_3\bar{b}_2 b_1\bar{b}_0$	2	SUB
0	0	1	1	$\bar{b}_3\bar{b}_2 b_1 b_0$	3	CAD
0	1	0	0	$\bar{b}_3 b_2\bar{b}_1\bar{b}_0$	4	CSB
0	1	0	1	$\bar{b}_3 b_2\bar{b}_1 b_0$	5	JPO
0	1	1	0	$\bar{b}_3 b_2 b_1\bar{b}_0$	6	JUN
0	1	1	1	$\bar{b}_3 b_2 b_1 b_0$	7	JNE
1	0	0	0	$b_3\bar{b}_2\bar{b}_1\bar{b}_0$	8	
1	0	0	1	$b_3\bar{b}_2\bar{b}_1 b_0$	9	
1	0	1	0	$b_3\bar{b}_2 b_1\bar{b}_0$	10	NOP
1	0	1	1	$b_3\bar{b}_2 b_1 b_0$	11	
1	1	0	0	$b_3 b_2\bar{b}_1\bar{b}_0$	12	
1	1	0	1	$b_3 b_2\bar{b}_1 b_0$	13	
1	1	1	0	$b_3 b_2 b_1\bar{b}_0$	14	STO
1	1	1	1	$b_3 b_2 b_1 b_0$	15	CMF

ตารางที่ ๓.๔ Functional Operation of Address and Function Decoders



รูปที่ ๓.๑๒ Logic Diagram of Address and Function Decoders

๓.๓.๔ Transfer Gate

ในการ transfer data จาก porgram register ไปยัง Address register, Address decoder และ Function decoder และจาก Address register ไปยัง Address decoder จำเป็นจะต้องมี transfer gate เป็นตัวควบคุมซึ่งมีทั้งหมด 4 gates ดังรูปที่ ๒.๖ transfer gate แต่ละตัวจะเปิดให้ data ผ่านขึ้นอยู่กับการควบคุมดังนี้ transfer gate 1 เปิดเมื่อมีคำสั่งเกี่ยวกับการคำนวณ (AC) หรือ transfer data จาก Accumulator ไปยัง Memory register ที่ต้องการ (STO) Transfer gate 2เปิด เมื่อมี Jump condition เกิดขึ้น (JI) Transfer gate 3 เปิดในขณะที่เครื่องอยู่ใน phase ของการทำงาน ($\bar{X}$) และ Transfer gate 4 เปิดในขณะที่เครื่องอยู่ใน phase ของการอ่านโปรแกรม (X) Transfer gate แต่ละตัวจะประกอบด้วย AND gate ๔ ตัวซึ่งใช้ transfer output และ $\overline{\text{output}}$ ของ address code, operation code และ Address register ทั้ง 4 bits ไปยัง decoder ที่ต้องการ ฟังค์ชั่นการทำงานและวงจรของ transfer gate จะมีลักษณะดังตารางที่ ๓.๖ และรูปที่ ๓.๑๓

B	C	T
0	0	0
0	1	0
1	0	0
1	1	1

Note : B = Binary input

C = Control input i .

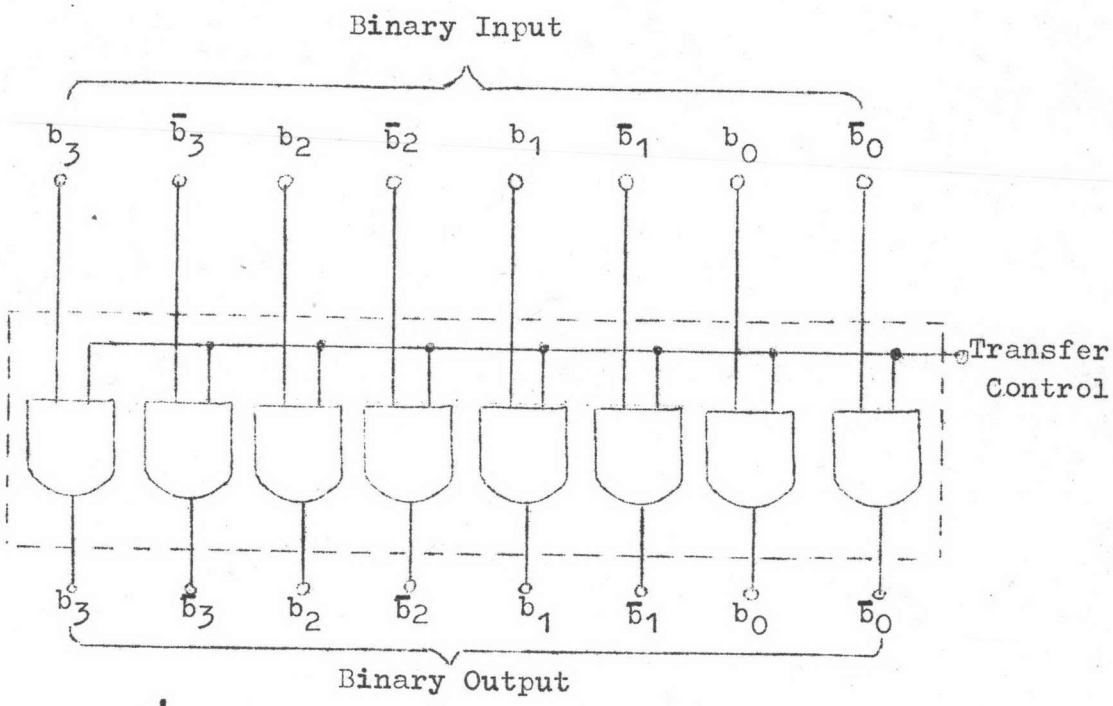
ie. AC + STO, JI, $\bar{X}$ or X

T = Transfer binary output

Function :

T = BC

ตารางที่ ๓.๖ Truth Table for each gate of Transfer Gate



รูปที่ ๓.๑๓ Logic Diagram of Transfer Gate

๓.๓.๕ Instruction Control Gate and Stop Indicator

ในช่วงเวลา execution คำสั่งจาก function decoder จะไปควบคุมส่วนต่าง ๆ ของเครื่องให้ทำงานตามที่ต้องการในตารางที่ ๒.๑ ซึ่งในบางส่วนของวงจรจะต้องทำงานตามคำสั่งหลายอย่าง ดังนั้นจึงจำเป็นต้องรวมคำสั่งที่เกิดขึ้นไปควบคุมส่วนนั้นซึ่งเรียกว่า instruction control gate หรือ instruction control generator ฟังค์ชั่นที่ใช้ในการควบคุมมีดังนี้

Arithmetic Clear Control Function (CA) ใช้สำหรับ clear Accumulator เมื่อมีคำสั่ง CAD หรือ CSB ซึ่งมีฟังค์ชั่นดังนี้

$$CA = CAD + CSB$$

Add and Subtract Control Function (AS) ใช้สำหรับควบคุมวงจร overflow checker เมื่อมีคำสั่ง ADD หรือ SUB เกิดขึ้น มีฟังค์ชั่นดังนี้

$$AS = ADD + SUB$$

Complement Transfer Gate Control Function (CT) ควบคุมให้ output ของ Buffer register เปลี่ยนเป็น one's complement ก่อนเข้าวงจร parallel full adder เมื่อมีคำสั่ง SUB หรือ CSB มีฟังค์ชั่นดังนี้

$$CT = SUB + CSB$$

Arithmetic Control Function (AC) ใช้ควบคุม input data transfer ของ Buffer register และ Accumulator เมื่อมีคำสั่งเกี่ยวกับการคำนวณทั้งหมดได้แก่ ADD, SUB, CAD หรือ CSB ซึ่งมีฟังค์ชั่นดังนี้

$$AC = ADD + SUB + CAD + CSB$$

$$= AS + CA$$

Transfer Gate 1 Control Function (T_1) ควบคุม transfer gate 1 ให้ address code ใน program register ผ่านไปยัง address decoder เมื่อมีคำสั่งเกี่ยวกับการคำนวณหรือ transfer data เข้าไปเก็บใน memory register ฟังค์ชั่นของวงจรคือ

$$T_1 = ADD + SUB + CAD + CSB + STO$$

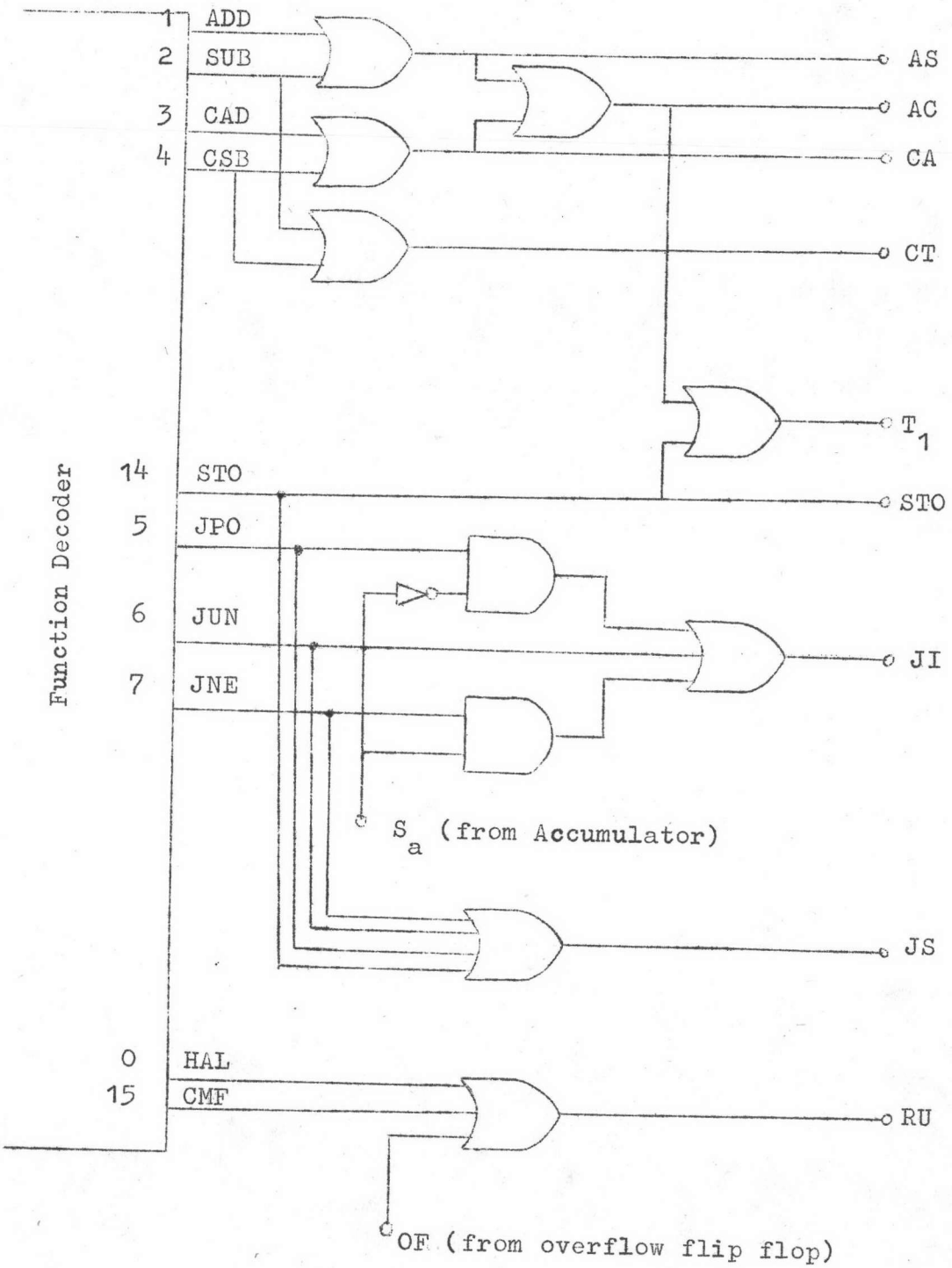
$$= AC + STO$$

Jump Instruction Control Function (JI) ใช้ควบคุมให้ Address register หยุดนับและ transfer address code ใน program register มายัง address register ด้วย transfer gate 2 เมื่อมีคำสั่ง JUN, JPO หรือ JNE สำหรับคำสั่ง JPO และ JNE จะเกิด Jump condition ได้ก็เมื่อ Sign bit (S_a) ใน Accumulator มีเครื่องหมายตรงตามคำสั่ง กล่าวคือ JPO เกิดขึ้นถ้า Sign bit เป็นบวก (0) และ JNE เกิดขึ้น ถ้า Sign bit เป็นลบ (1) ซึ่งฟังค์ชั่นของ JI จะเขียนได้ดังนี้

$$JI = JUN + \bar{S}_a \cdot JPO + S_a \cdot JNE$$

Automatic Stop Control Function (RU) ควบคุมให้เครื่องหยุดทำงานเมื่อมีคำสั่ง HAL, CMF หรือเกิด overflow (OF) ฟังค์ชั่นของวงจรคือ

$$RU = HAL + CMF + OF$$



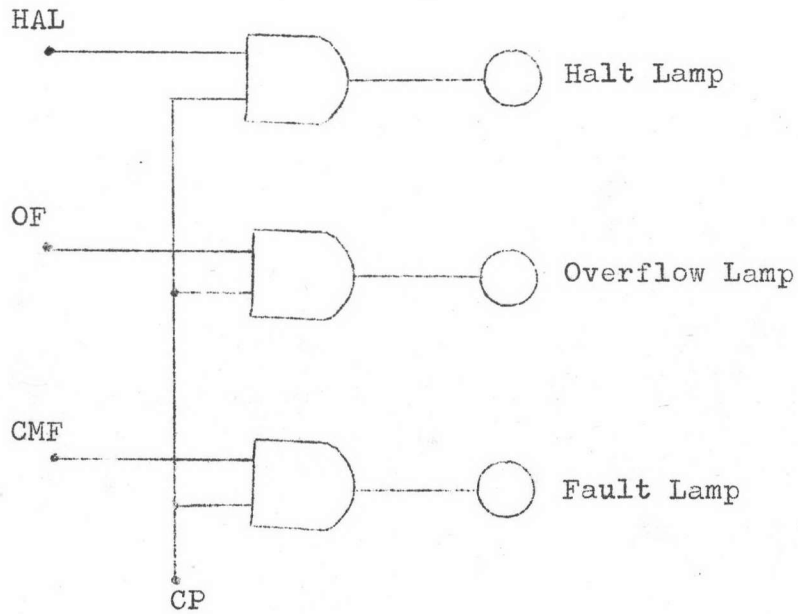
รูปที่ ๓.๑๔ Logic Diagram of Instruction Control Gate

Jump Sequence Control Function (JS)ใช้ควบคุมให้เครื่อง Jump เวลาการทำงานใน phase Z ในกรณีที่การทำงานของเครื่องในรอบนั้นต้องการเพียง phase X และ phase Y เท่านั้นซึ่งได้แก่การทำงานของคำสั่ง JPO, JUN, JNE หรือ STO ฟังค์ชั่นของ JS คือ

$$JS = JPO + JUN + JNE + STO$$

สำหรับการควบคุมหน่วยความจำที่เกี่ยวกับคำสั่งจะมีอยู่อย่างเดียวคือคำสั่ง STO ซึ่งจะต่อโดยจาก Function decoder ผ่านวงจร driver ไปยัง Memory register ทั้ง 16 wordsวงจรของ Instruction control gate ที่กล่าวมาทั้งหมดนี้จะมีลักษณะดังรูปที่ ๓.๑๔

ส่วนวงจร Stop Indicator จะต่อมาจากคำสั่งที่ทำให้เครื่องหยุดทำงานคือ HAL, CMF และ OF ซึ่งต่อร่วมกับ clock pulse (CP) แล้วต่อเข้ากับหลอดไฟเพื่อแสดงผลโดยที่หลอดไฟจะกระพริบตามจังหวะของ CP ตลอดเวลาถ้ามีคำสั่ง HAL, CMF หรือเกิด overflow ขึ้น วงจรของ Stop Indicator จะมีลักษณะดังรูปที่ ๓.๑๕

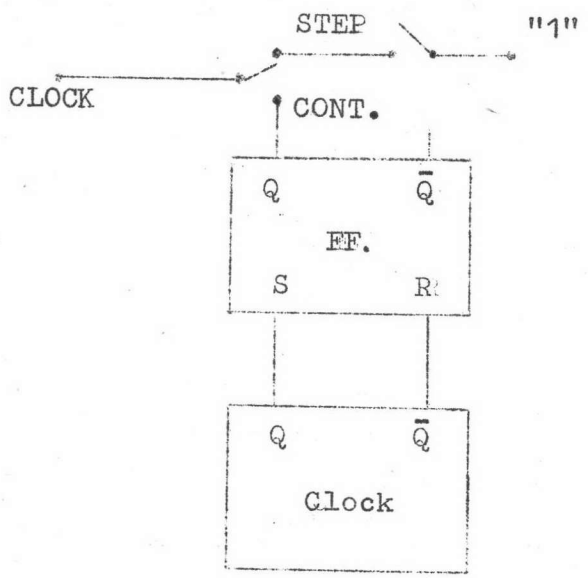


รูปที่ ๓.๑๕ Logic Diagram of Stop Indicator

๓.๔ หน่วยควบคุมลำดับการทำงาน

๓.๔.๑ Clock Generator

วงจร clock ประกอบด้วยวงจร square wave generator ซึ่งใช้ Astable multivibrator และ RS flip-flop ต่อร่วมกันและมีสวิทช์สำหรับให้เครื่องทำงานแบบ continuous หรือ step ก็ได้ดังรูปที่ ๓.๖

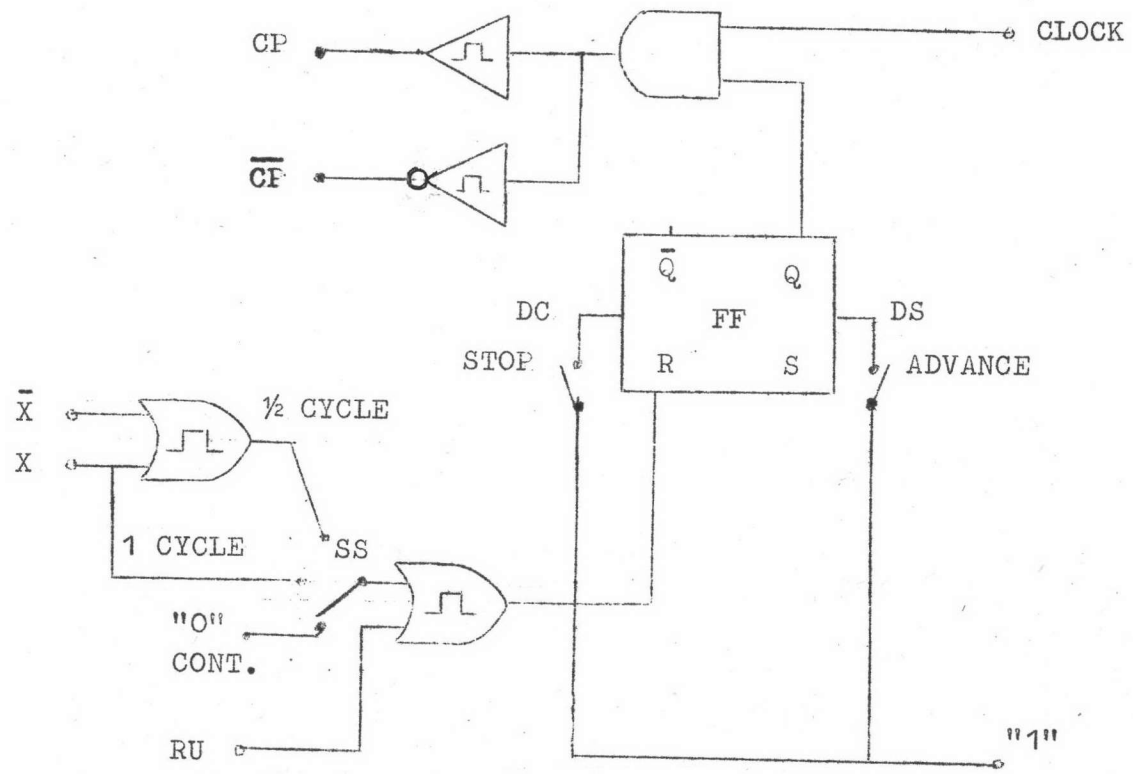


รูปที่ ๓.๑๖ Logic Diagram of Clock Generator

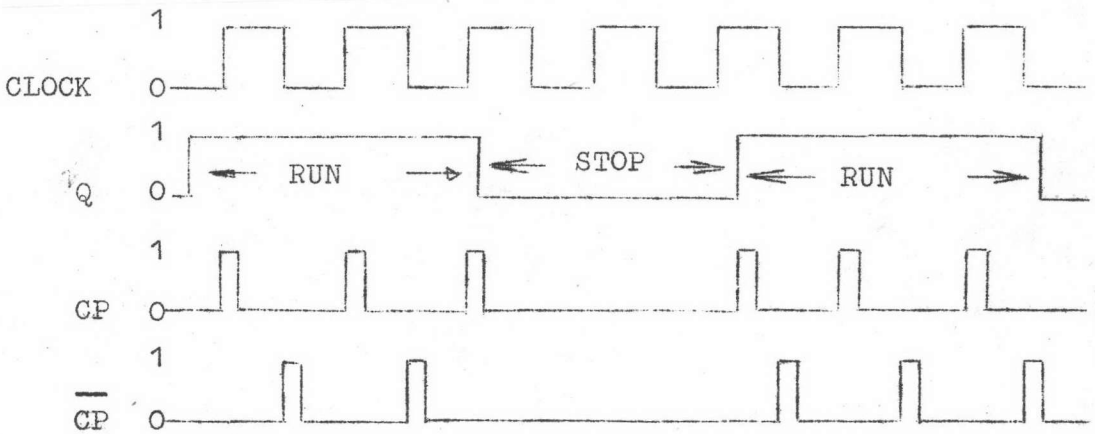
๓.๔.๒ Timing Control Circuit

วงจร Timing control ใช้สำหรับควบคุมการ start และ stop เครื่อง โดยใช้ RS flip-flop ดังรูปที่ ๓.๑๗ CP จะผ่านไปยังระบบต่าง ๆ ของเครื่องได้เมื่อ output ของ flip-flop เป็น "1" โดยการกดสวิทช์ Advance DS ซึ่งเครื่องจะเริ่มต้นการทำงาน และเครื่องจะหยุดทำงานเมื่อกดสวิทช์ Stop DC หรือ flip-flop ถูก clear ด้วยผลของคำสั่งที่ทำให้เกิด RU function เป็น "1" นอกจากนี้

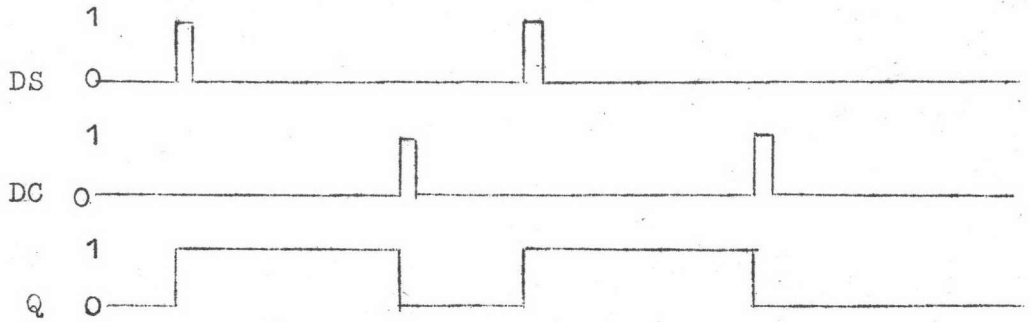
เครื่องสามารถหยุดทำงานเมื่อครบ ๑ รอบ หรือครึ่งรอบได้โดยใช้ Selector Switch (SS) เป็นตัวเลือกรับสัญญาณจากเครื่องมา clear flip-flop ซึ่งสำหรับ ๑ รอบ เครื่องจะหยุดทำงานทุกครั้งเมื่อเริ่มต้น program time ของรอบต่อไป และสำหรับครึ่งรอบเครื่องจะหยุดทำงานทุกครั้งเมื่อเริ่มต้น program time และเริ่มต้น execution time ซึ่ง waveform ของการควบคุมจะเป็นไปดังรูปที่ ๓.๑๘



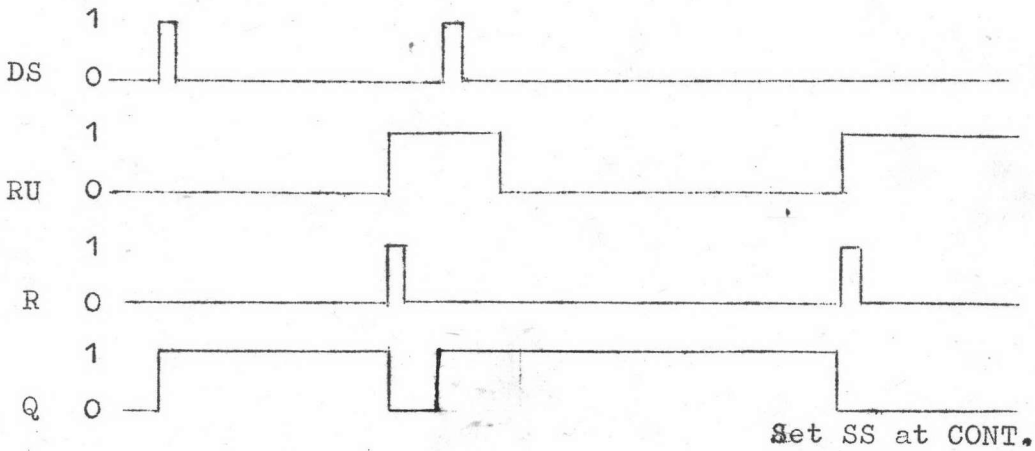
รูปที่ ๓.๑๗ Logic Diagram of Timing Control Circuit



ก) Waveforms of CP and $\overline{CP}$ output

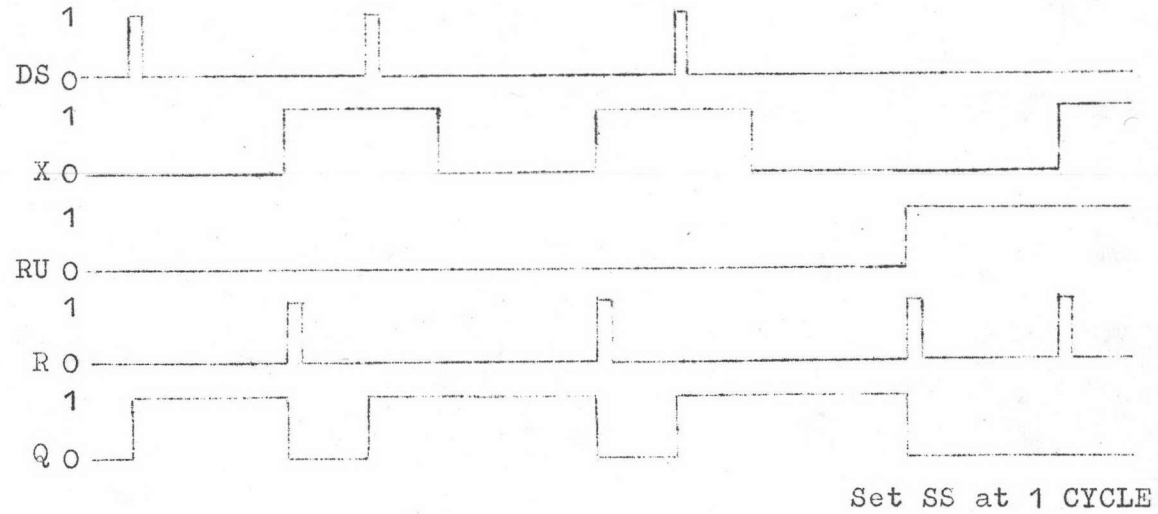


ข) Waveform of Q under DS and DC

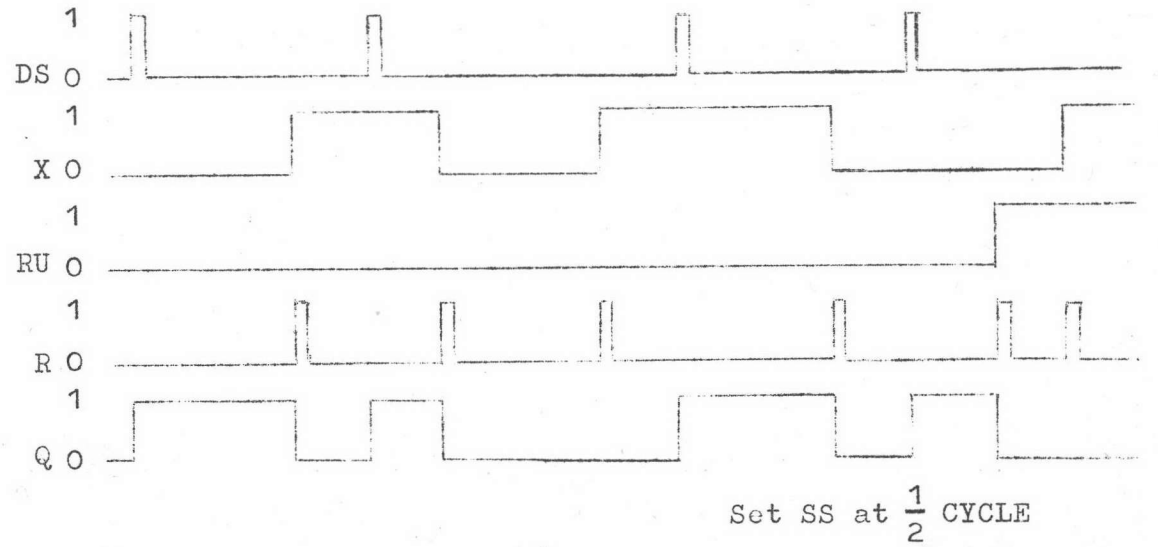


ค) Waveform of Q under Continous Run condition

รูปที่ ๓.๑๘ Waveforms of Timing Control at Various Condition



ง) Waveform of Q under 1-Cycle Run Condition



จ) Waveform of Q under $\frac{1}{2}$ Cycle Run Condition

รูปที่ ๓.๑๘ Waveforms of Timing Control at Various Condition (cont.)

๓.๔.๓ Sequence Control Circuit

วงจร Sequence control ประกอบด้วย sychronous 2 bit counter โดยใช้ clocked-RS flip flop ซึ่งทำเป็นวงจร Divide-by-2 และ Divide-

by-3 counter ซึ่งควบคุมด้วย JS function และ output ของวงจรจะ decode ให้ได้สัญญาณของ phase X, Y และ Z ตามลำดับลักษณะการทำงานของวงจรจะเป็นดังตารางที่ ๓.๗ และรูปที่ ๓.๑๙ และ ๓.๒๐ (พิจารณาภาคผนวก ข ประกอบ) การนับของวงจรจะเปลี่ยนเมื่อได้รับสัญญาณ $\overline{CP}$ จากวงจร Timing control ทั้งนี้เพื่อให้เวลาของสัญญาณ CP อยู่ระหว่างกลางของแต่ละ phase ตามรูปที่ ๒.๒

Phase	Output of Counter	
	JS=0	JS=1
	Q_1Q_0	Q_1Q_0
X	1 0	1 0
Y	0 1	0 1
Z	0 0	-

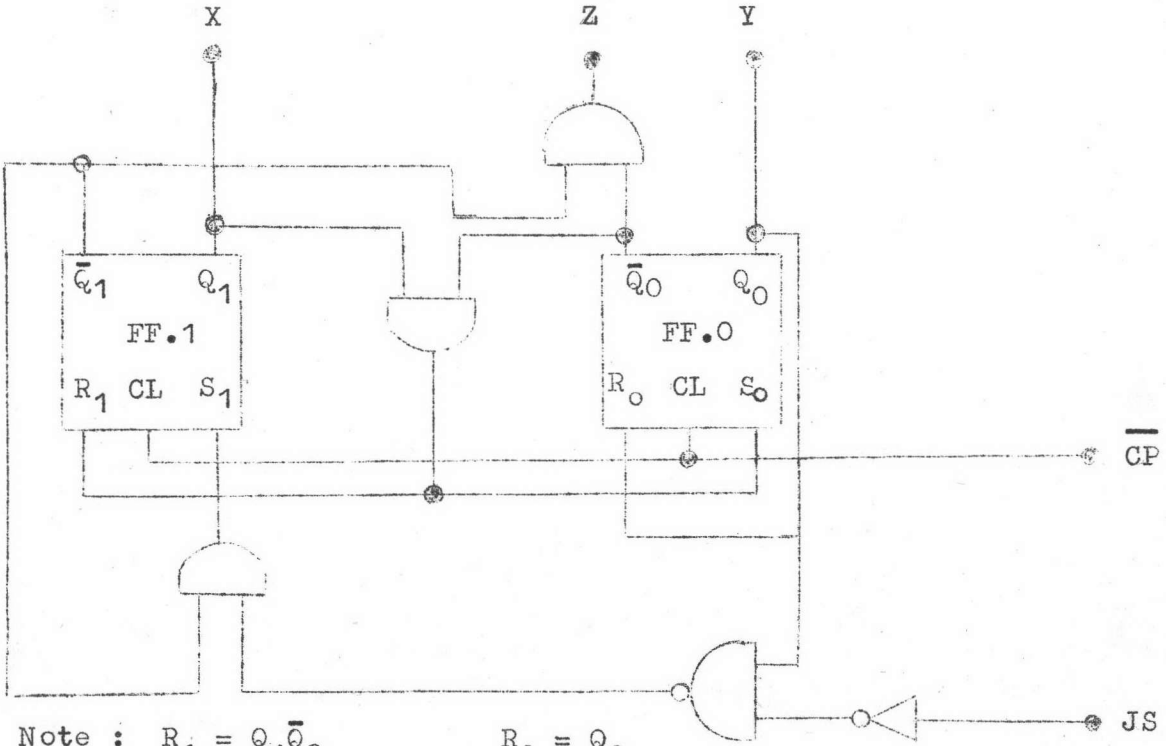
Note : Q_1 and Q_0 = output of FF1 and FF0 respectively

Function; $X = Q_1$

$Y = Q_0$

$Z = \bar{Q}_1\bar{Q}_0$

ตารางที่ ๓.๗ Count Table of Sequence Control

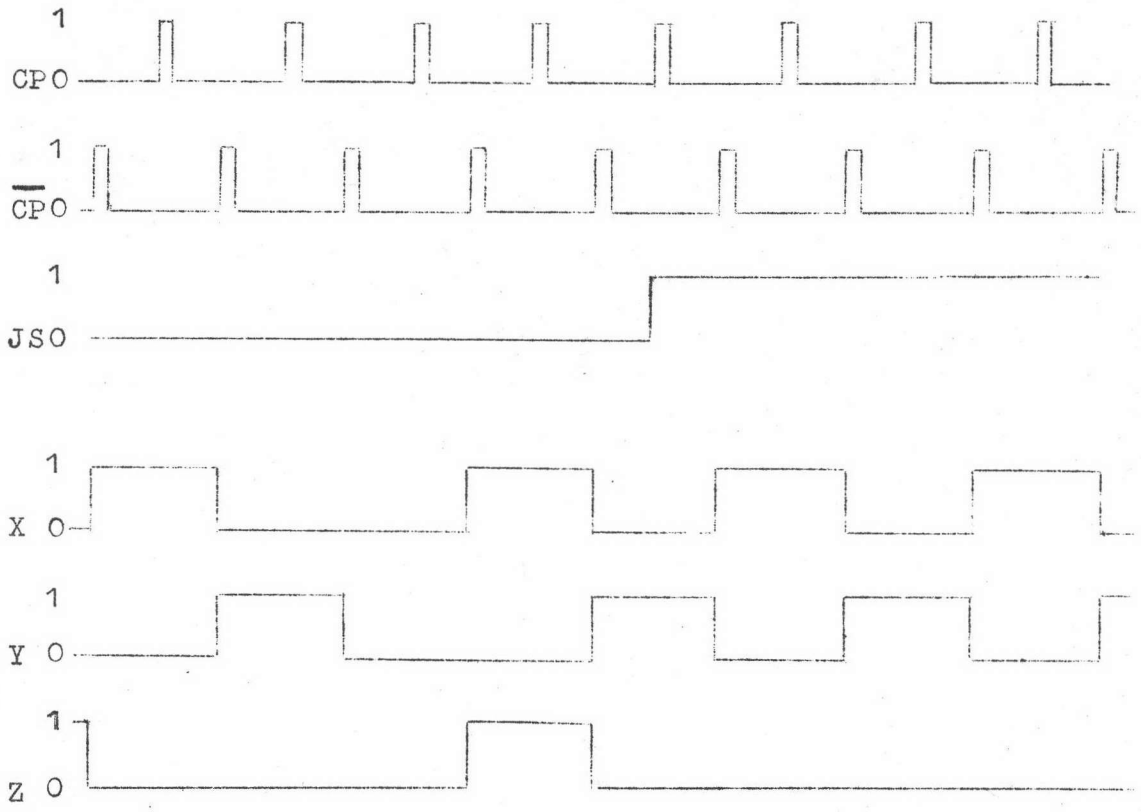


Note : $R_1 = Q_1\bar{Q}_0$ $\quad R_0 = Q_0$

$S_1 = \bar{Q}_1 \cdot (\overline{JS \cdot Q_0})$ $\quad S_0 = Q_1\bar{Q}_0$

ดูการออกแบบวงจรในภาคผนวก ข

รูปที่ ๓.๑๙ Logic Diagram of Sequence Control Circuit



รูปที่ ๓.๒๐ Waveforms of Sequence Control Circuit

.....