Booth's Algorithm

* If we want to Multiple two integer values so the system apply some kinds of algorithm is called booth's algorithm.

* Accumulator :- always initialize by zeroes
  Multiplicant :- Multiply with [M]
  Multiplier :- Multiply by [Q]
  Q-1 :- Always initialized by zero

Ex 1:- Multiplying 2 +ve Integers using Booth's Algorithm
7 x 3

Sol $\underset{M}{7} \times \underset{Q}{3}$ M = 0111 and -M = 1001
Q = 0011

If 0→1 Sub
1→0 Add
0-0 } No
0→1 } chan

	Acc	Q	Q-1	M	Operation
	0000	0011	0	0111	Sub 0000 + 1001 1001
①					
R.S	1001	0011			
	1100	1001	1		
②					Right Shift
R.S	1110	0100	1		
③	0101	0100			Add 1110 + 0111 0101
R.S	0010	1010	0		
4 R.S	0001	0101	0		

* we have to stop our process when we complete our 4 steps

No. of bits = No. of Steps

∴ 0001 0101 = 21 Ans

Ex 6 × 3 registered into 4 bit
↓ ↓
M Q

M = 0101
Q = 0011
-M = 1011

	Acc	Q	Q_{-1}	M	operations
	0000	0011	0	0101	
①					sub 0000 + 1011 1011
R.S	1011	0011			
	1100	1001	1		
② R.S	1110	1100	1		only shift
③	0011	1100			Add 1110 + 0101 0011
R.S	0001	1110	0		
R.S	0000	1111	0		

wrong one

Ex -7×3 register into 5 bits
$\underset{M}{-7} \times \underset{Q}{3}$

M = (−7) → 0111 → 1001 → 011
Q = 3 → 0011 →
−M = 0111

Acc into 5 bits
−M = 00111 and M = 11001
Q = 00011

	Acc	Q	Q−1	M	operation
	00000	00011	0	11001	sub
①					00000 + 00111 = 00111
R.S	~~00111~~	00011			
	00011	10001	1		
② Rs	00001	11000	1		Only Right shift
③	110~~1~~0	11000			Add 00001 + 11001 = 11010
R.S	11101	01100	0		
④ R.S	11110	10110	0		Right shift only
⑤ R.S	11111	01011	0		Right shift only

As it is 1111101011 we have to takes its 2's comp
∴ 0000010101 = −21 Ans

Que Solve -7×-3 in 4 bits booths algorithm

Sol $\underset{M}{\underset{\downarrow}{-7}} \times \underset{Q}{\underset{\downarrow}{-3}}$

$M = (-7) = 0111 \rightarrow 1001$

$Q = (-3) \rightarrow 0011 \rightarrow 1101$

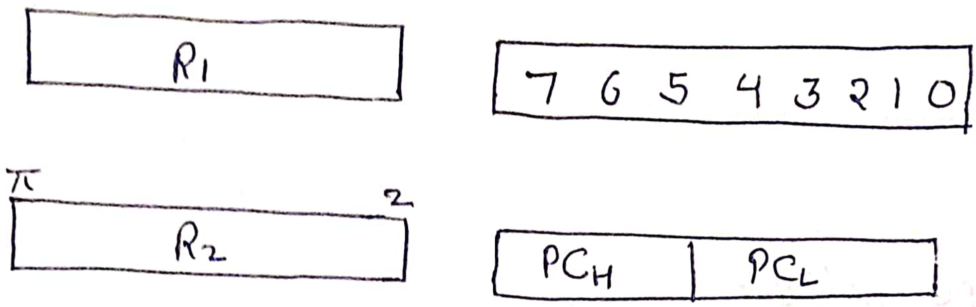
$-M = 0111$

	Acc	Q	Q-1	M	Operation
①	0000	1101	0	1001	Sub 0000 +0111 0111
R.S	0111	1101			
	0011	1110	1		
②	1100	1110			Add 0011 +1001 1100
R.S	1110	0111	0		
③	0101	0111			Sub 1110 +0111 0101
R.S	0010	1011	1		
④	0001	0101	1		Right Shift only

∴ 00010101 → +21 Ans

* We don't have to takes 2's complement

Register Transfer



If (P=1) then $R_2 \leftarrow R_1$

$P: R_2 \leftarrow R_1$

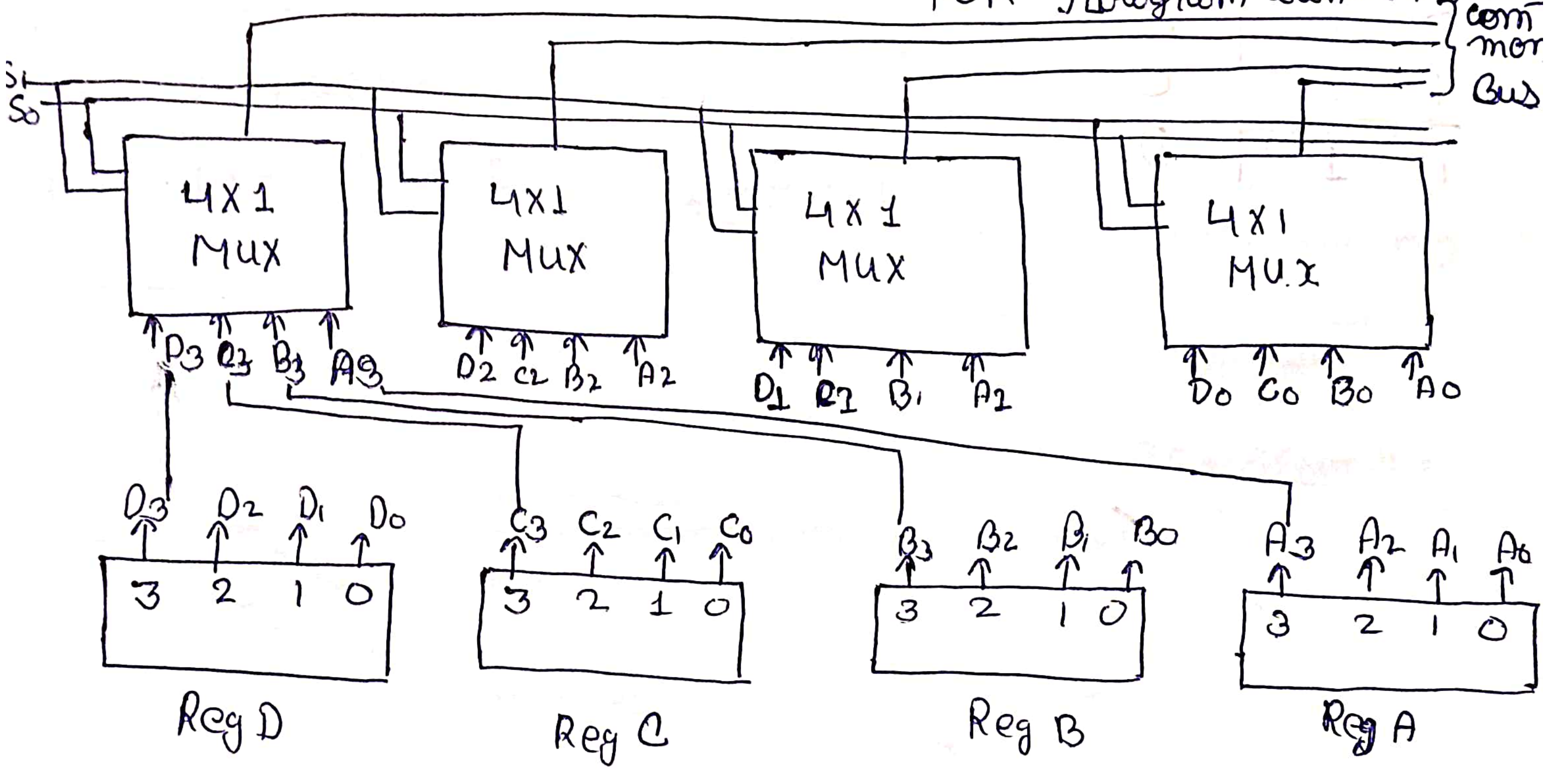
$T= R_2 - R_1 ,\ R_1 \leftarrow R_2$

MAR [Memory Address Register]

GPR [General Purpose Register]

IPR [Instruction Pointer Register]

PCR [Program Counter Register]



* Processesor have some memory
* PCR → Holds the Address of next memory location.
* Program Run as

Fetch
↓
Decode
↓
Execute

Q *GPR →
S

$PC_H \rightarrow$ Pc for upper Half bit
$PC_L \rightarrow$ Pc for lower Half bit

R

S_0	S_1	Register
0	0	A
0	1	B
1	0	C
1	1	D

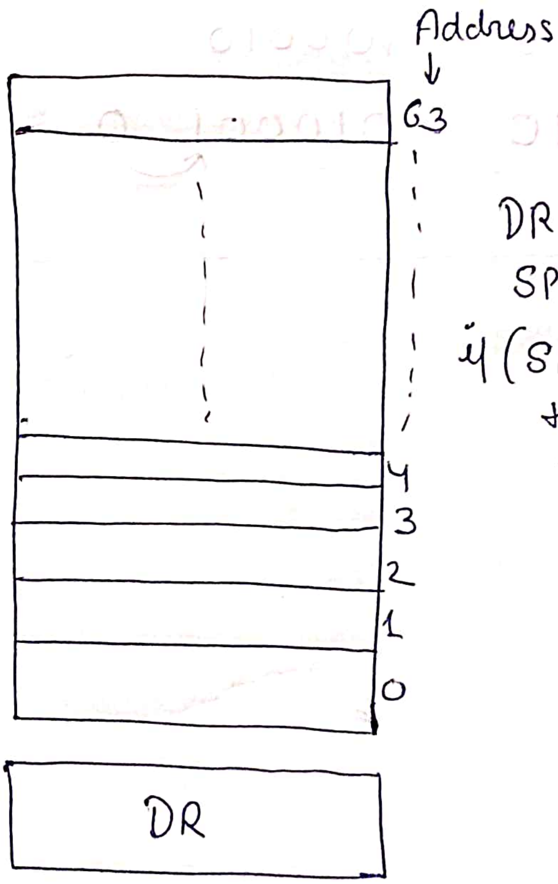
How to design any Circuits using MUX

* These are functionally complete.
* If we create any MUX using AND, OR, NOT gate using 4x1/2x1 then we can design any circuit

Register Stack Organisation

FULL

Empty



$SP \leftarrow SP+1$

$M[SP] \leftarrow DR$

if $(SP = 0)$ then

Full ← 1

Empty ← 0

SP →

$DR \leftarrow M[SP]$

$SP \leftarrow SP-1$

if $(SP=0)$

then

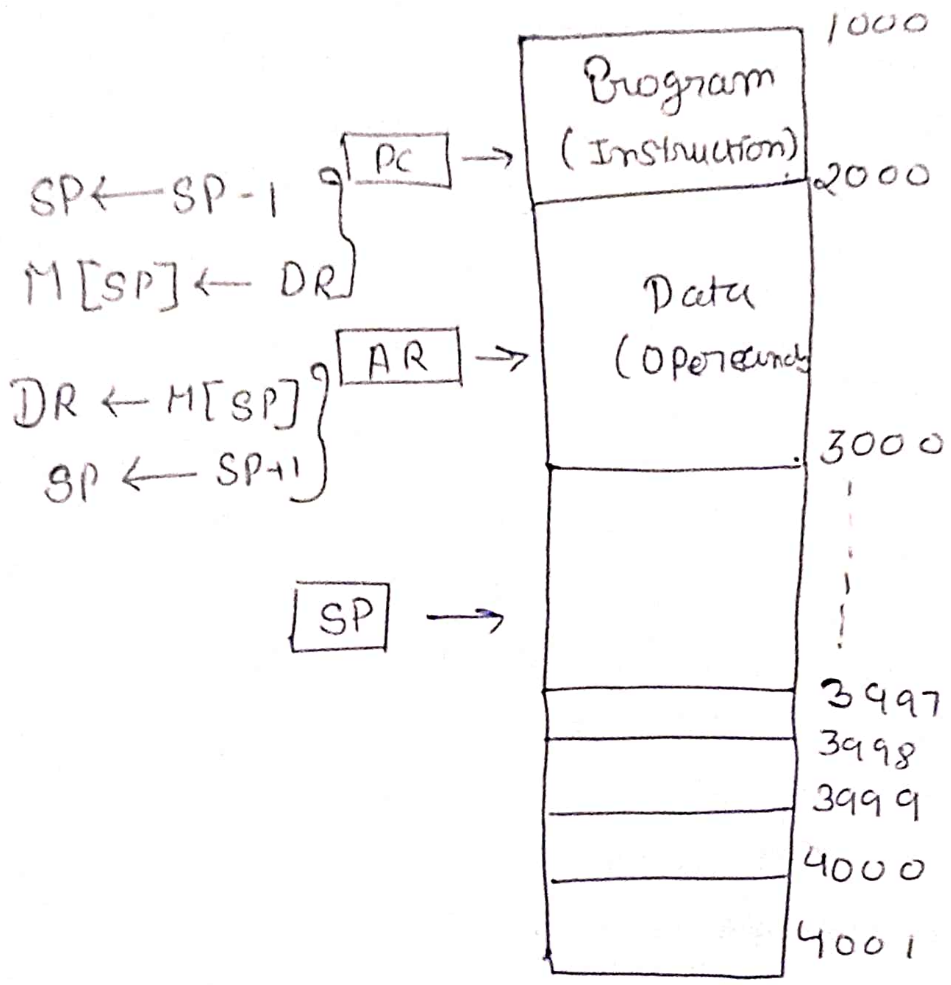
EMPTY ← 1

Full ← 0

* SP → Stack Pointer. Point at the top of Stack

* Push → Insert the data
* Pop → delete and taken data
* Full → [Stack memory is full] Full → 1 [

Memory Stack Organisation



PC → Hold the address of next Instruction
AR → " " " " Data.
SP → Hold the address of Stack address