gradeup

Prep Smart. Score Better.

Computer Organization and Architecture

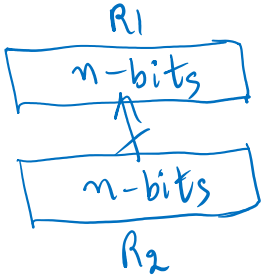
Register Transfer and Micro-operations

ABOUT ME : MURALIKRISHNA BUKKASAMUDRAM

- MTech with 20 years of Experience in Teaching GATE and Engineering colleges
- IIT NPTEL Course topper in Theory of computation with 96 %
- IGIP Certified (Certification on International Engineering educator)
- GATE Qualified
- Trained more than 50 Thousand students across the country
- Area of Expertise : TOC,OS,COA,CN,DLD

Register Transfer and Micro-operations

RTL (Register Transfer Language)

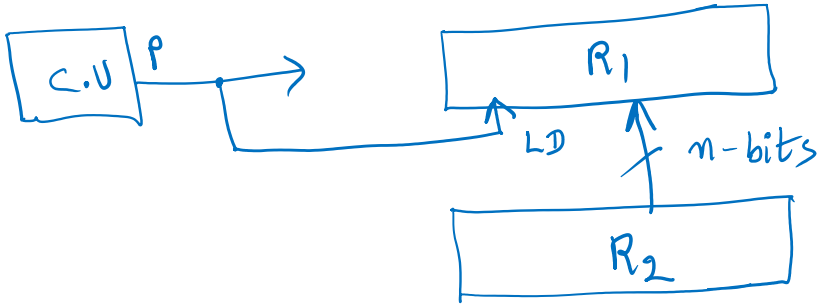


P : $R_1 \leftarrow R_2$

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

(LD = 1)

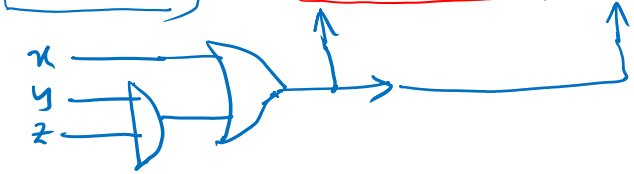
(LD = 0)



If (P = 0) then LD = 0

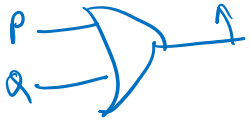
If (P = 1) then LD = 1

$x + yz$: $R_1 \leftarrow R_2$, $R_3 \leftarrow R_4$



Register Transfer and Micro-operations

P + Q : $R_1 \leftarrow R_2 + R_3$



T_1 : AC(0-7) ← INPR

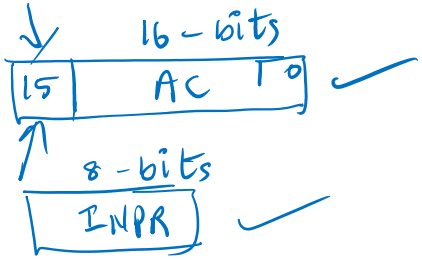
If ($T_1 = 1$) then

AC(15)

T_2 : OUTR ← AC(8-15)

: ←

; ( )



AC(15) = 0

AC(15) = 1

Micro-Operation

→ the operation, we can perform on the data stored in the register.

Register Transfer and Micro-operations

Example 1 :- If (P=1) then ($R_1 \leftarrow R_2$) else if (Q=1) then ($R_3 \leftarrow R_4$)

Sol :

P : $R_1 \leftarrow R_2$

$\bar{P}Q$: $R_3 \leftarrow R_4$

Memory Read

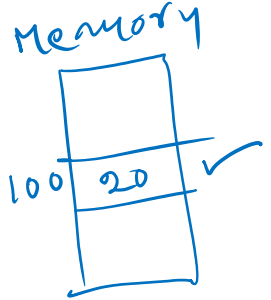
Read : $DR \leftarrow M[AR]$

Memory Write

Write : $M[AR] \leftarrow DR$

DR : Data Register

AR : Address Register



Register Transfer and Micro-operations

Micro - Operations

(1) Arithematic
(2) Logic ✓
(3) Shift ✓

- Addition → (A + B)
- Subtraction → (A − B)
- Increment → (A + 1) ✓
- Decrement → (A − 1) ✓

Addition ✓

HA FA

$A = A_3 A_2 A_1 A_0$
$B = B_3 B_2 B_1 B_0$
$C_3\ C_2\ C_1\ 0$ ←
C_4 $S_3\ S_2\ S_1\ S_0$

two n-bit Registers

1 HA + (n-1) FA's.

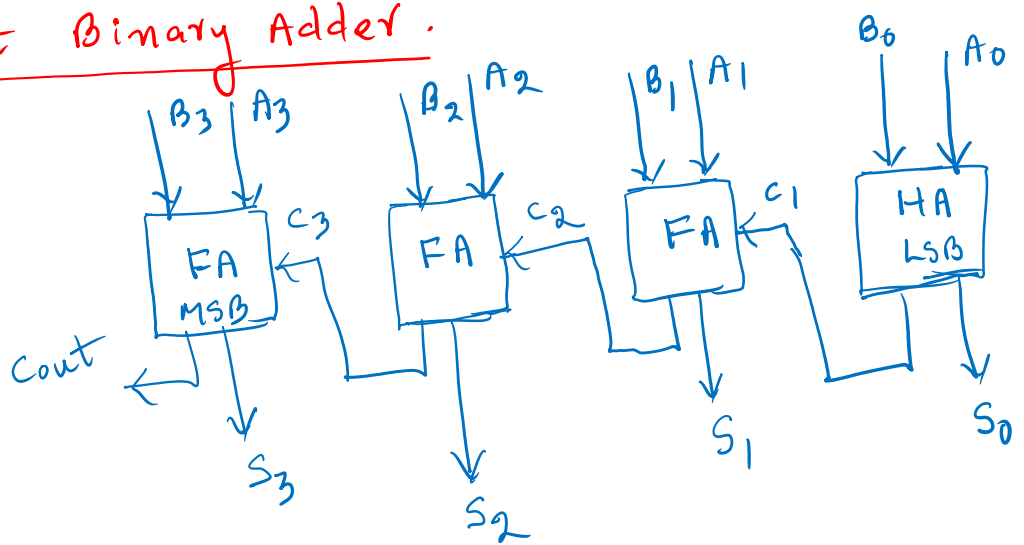
(OR)

n - FULL Adders

Register Transfer and Micro-operations

A + B

4-bit Binary Adder.



2 (10)

Example :-

A = 1001
B = 0011
0 1 1 ✓

0 1100

$S_3S_2S_1S_0$

Register Transfer and Micro-operations

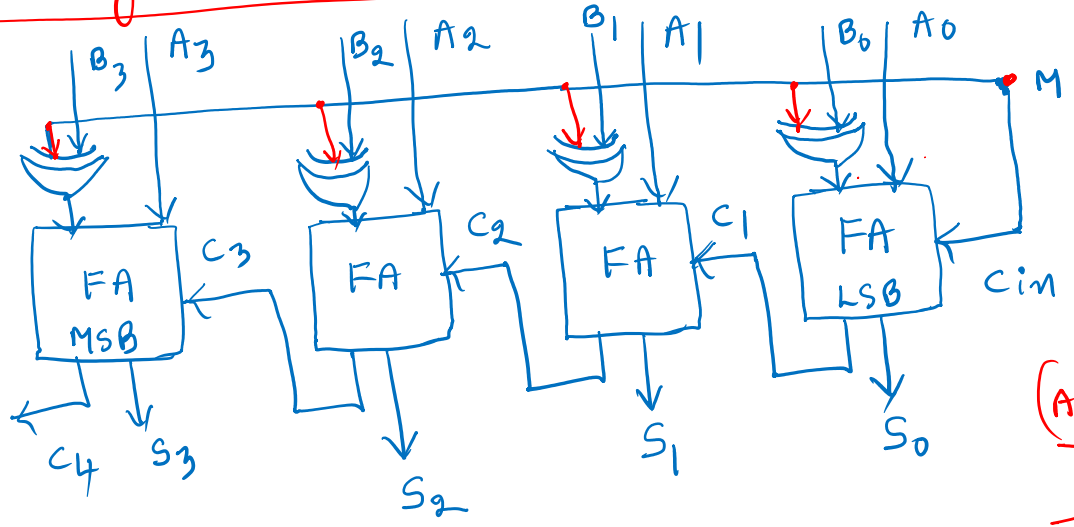
$A+B+0 =$

$A+B$

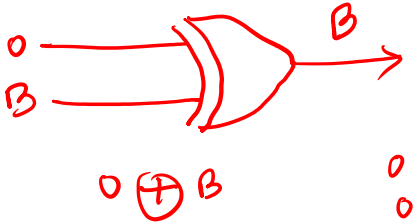
$A-B = A+(\bar{B}+1)$

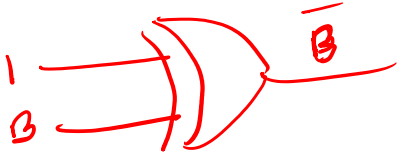
$M=0$ (Adder)
$M=1$ (Subtractor)

4-bit binary Adder Subtractor Circuit



$(A+\bar{B}+1)$





$0 \oplus 0 = 0$
$0 \oplus 1 = 1$

$1 \oplus 0 = 1$
$1 \oplus 1 = 0$

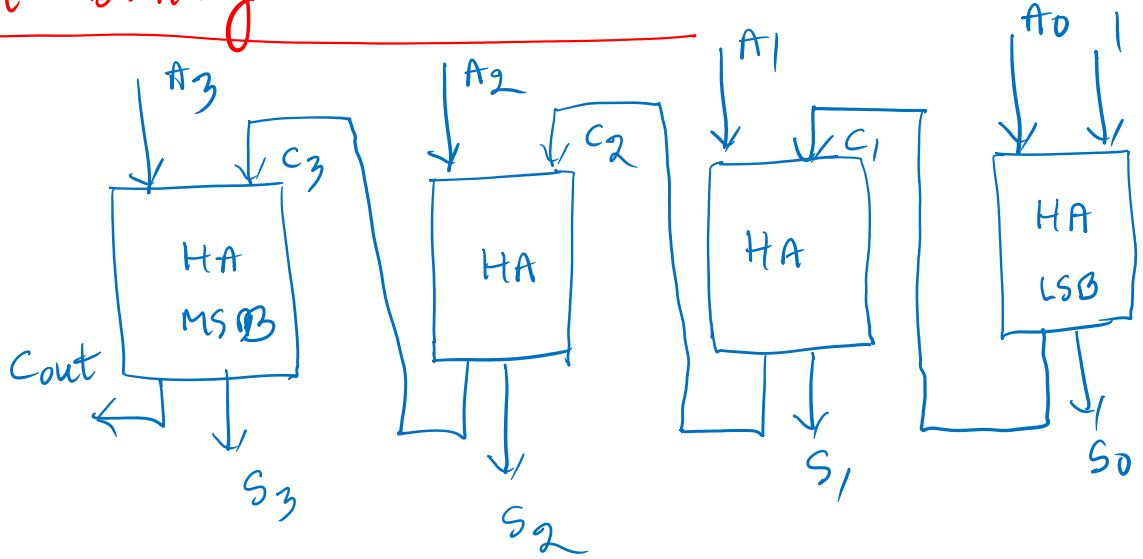
4-bit Binary Incrementer

A+1

$A = A_3 A_2 A_1 A_0$

$C_3\ C_2\ C_1\ 1$ ✓

C_4 $S_3\ S_2\ S_1\ S_0$



Register Transfer and Micro-operations

4-bit Binary Decrementor

A − 1

A + 2's Complement of 1.

$A = A_3 A_2 A_1 A_0$

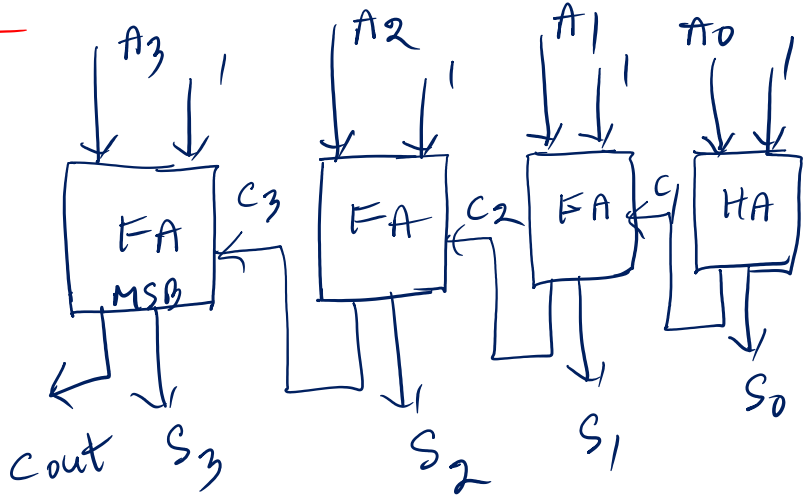
0 0 0 1

0001
1110
1111

1110
1
1111

A = 1010
1111
1 1001

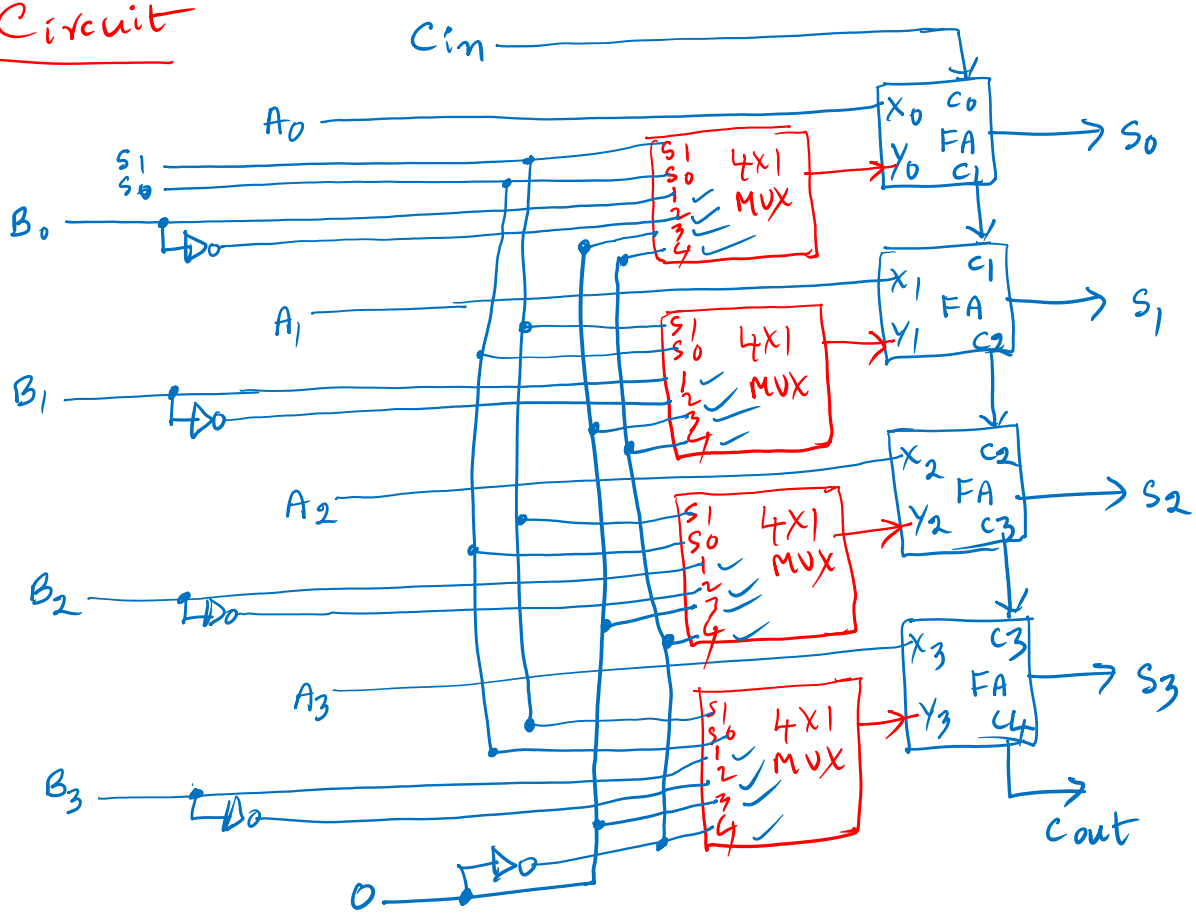
00000001
1111 1110
1
1111 1111



Register Transfer and Micro-operations

4-bit Arithematic Circuit

- A+B
- A-B
- A+1
- A-1



Register Transfer and Micro-operations

ADD
ADC

S_1 S_0 C_{in}	operand (Y)	$S = A + Y + C_{in}$	Micro Operation
0 0 0	B	$S = A + B$	Addition
0 0 1	B	$S = A + B + 1$	Addition with Carry
0 1 0	$\overline{B}$	$S = A + \overline{B}$	Barrow Subtraction
0 1 1	$\overline{B}$	$S = A + \overline{B} + 1$	Subtraction.
1 0 0	0's	$S = A$	Transfer
1 0 1	0's	$S = A + 1$	Increment
1 1 0	1's	$S = A - 1$	Decrement
1 1 1	1's	$S = A - 1 + 1$ A	Transfer

Register Transfer and Micro-operations

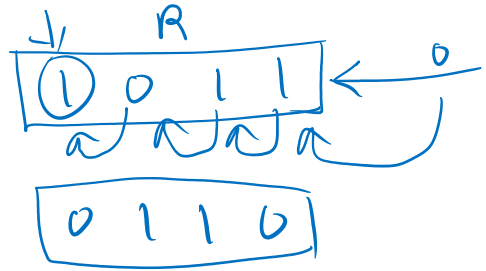
Shift Micro-Operations

(1) Logical shift → lshl, lshr

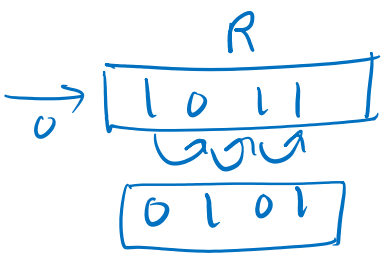
(2) Circular shift → cil, cir

(3) Arithematic Shift → ashl, ashr.

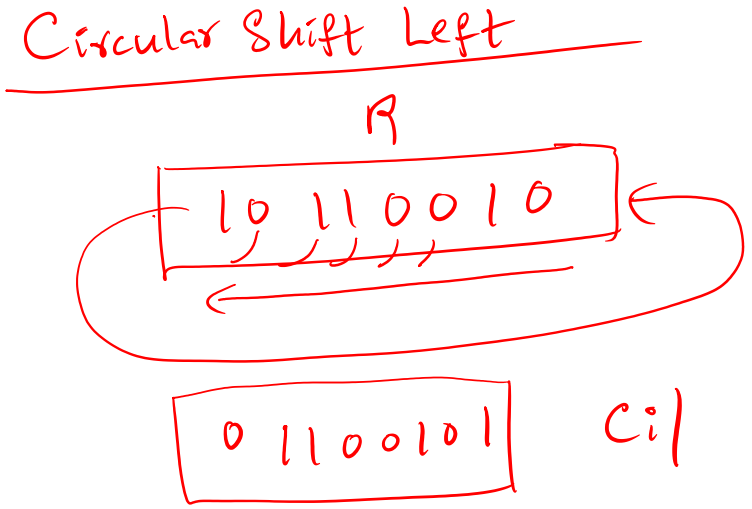
Logical shift Left

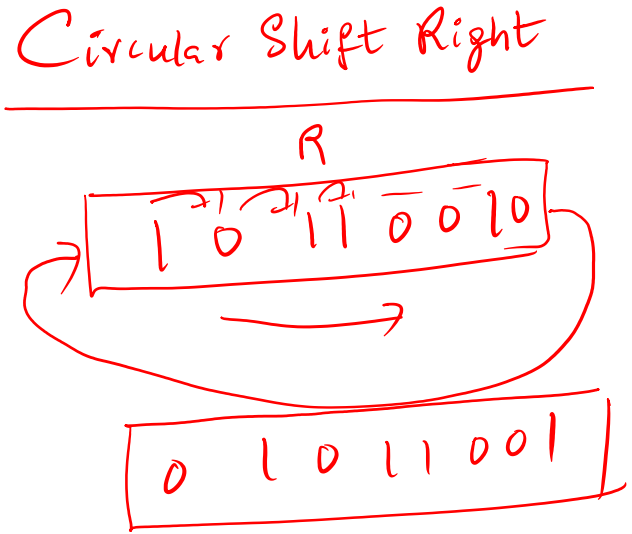


Logical Shift Right



Register Transfer and Micro-operations





Register Transfer and Micro-operations

Arithematic Shift Left (ashl)

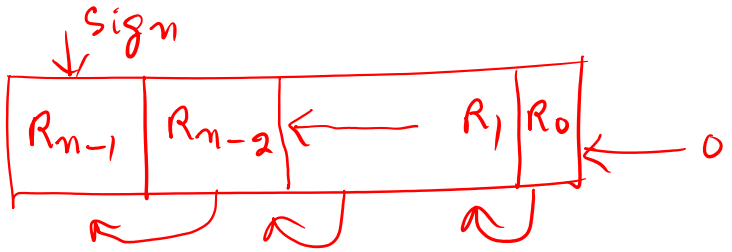
this multiply the given signed integer by 2.

+5 x 2 = +10

0 0101 → 0 1010

0 ⊕ 0 = 0

+10



$V_s = R_{n-1} \oplus R_{n-2}$

+10 x 2 = +20

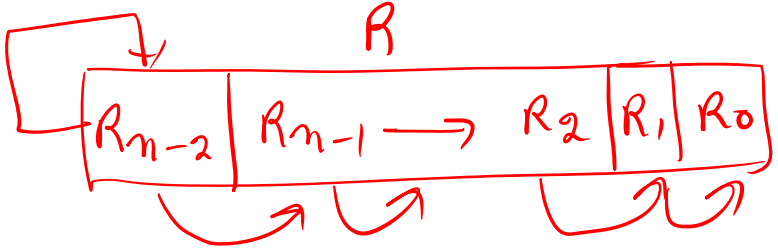
0 1010 → 1 0100

0 ⊕ 1 = 1

Register Transfer and Micro-operations

Divides the given Signed integer by 2

Arithematic Shift Right (ashr)



+5/2 = 2

0 0101

0 0010

+10

0 1010

0 0101

+10/2 = +5

+2

-10

1 0110

1 1011

1 1010

0101

1

0110