- **Macro programming**

Computer has got internal bus (ib).

Registers → A, B, C, D, PC, (S, T)

PC → Program Counter

(S, T) ← carry out arithmetic operations

We can put data from register to internal bus and vice-versa.

ALU - Arithematic Logic Unit

S & t are special registers

S and t can take value of ib but ib cannot

- **Gates**

and (.) , or (+)

$5a+7b >= 3$ — $a=1$ or $b=1$ — $(a+b)$

$8a+3b >= 10$ — $a=1$ and $b=1$ — $a.b$

$8a+3b >= 5$ — $a=1$ — $a$

$3a+7b+2c >= 8$ — $b=1$ and $a=1$ or $c=1$ — $b.(a+c)$

$7a+4b+3c >= 6$ — $a=1$ or ($b=1$ and $c=1$) — $a+b.c$

$7a+4b+3c+20d >= 26$ — $d=1$ — $d.(a+b.c)$

$3a+7b+2c+20d >= 8$ — $d=1$ or $d=0$ — $d+b.(a+c)$

$7a+4b+3c+20d+22e >= 26$ — $e=0$ $d.(a+b.c)$ or $e=1$ ( $d.(a+b.c)+e.(a+b+d)$ )

- $a+b'.c$ write inequality??

→ $2a+c-b >= 0$

- $a.b+c.d$

$ab+cd >= 1$

not (')

$K+K'.g = K+g$

$5a-7b >= -4$ — $a=1$ or $a=0$ and $b=0$ — $a+b'$

$7a+4b-3c+20d >= 23$ — $d=1$ and $(a+b.c')$

$7a+4b-3c >= 3$ — $d.(a+b.c')$

- $\boxed{abcd > pqrs}$

$a=1 \overset{ab>pq}{\text{and}} p=0 \rightarrow a.p'$

$a=1$ and $p=1$ ,

$a=0$ and $p=0$

$(a.p + a'.p').b.q' + a.p' \rightarrow ab > pq$

$ab > pq \rightarrow a > p$ or $a == p$ and $b > q$

$ab > pq \rightarrow a > p$ or $a >= p$ and $b > q$

$a.p' + (a+p').b.q'$

$a >= p$ when $a < p$ is false

$(a'.p)' = a + p'$

- $abc > pqr$

$ab > pq$ or $\underset{a=p \text{ and } b=q}{ab == pq}$ and $c > r$

$a.p' + (a+p').bq' + (a.p + a'p').(b'q' + bq).c.r'$

$a.p' + (a+p').[b.q' + (b + q').cr']$

$ab >= pq$ when $ab < pq$ is false

$(p.a' + (p + a').q.b')'$

$= (a+p').(b+q' + p'.q).cr'$

$a.p' + (a+p').[(b+q'+p'.q).cr' + bq']$

$= a.p' + (a+p').[(b+q').c.r' + b.q']$

$(a > p)$ or $(a >= p)$ and $bc > qr$

$a.p' + (a+p').[b.q' + (b+q').cr']$

- Combinatorial circuits

$P = (x \text{ and } y) \text{ or } z'$    output = $p$   input = $x, y, z$

↳ This is combinatorial circuit

$\langle xyz \rangle$ = 000, 010, 100, 110, 111 ⟶ true

$P = (x \text{ or } z') \text{ and } (y \text{ or } z)$ → 110, 111, 101, 010

input $x, y, z$   output $q$

$q = x \cdot z' + y \cdot z$ → 011, 100, 110, 111

$q = q \cdot z' + y \cdot z$    input $y, z$    output $q$

$yz = 11$, $q = 1$

$yz = 00$   retain

$yz = 01$, $q = 0$    $(q \cdot 0 + 0 \cdot 1)$

$yz = 10$   retain

$yz$ is made 11 output is 1.

Now $yz$ is changed to 10 output remains 1.

new $q$ = (old $q$) $\cdot z' + y \cdot z$

$yz$ is made 01, output = 0

Faults

and : · x.y    xy 11 -> 1

xy 10 -> 0

xy 00 -> 0    xy 01 -> 0

Input y stuck at 0

let x = 0, y = 1 input is given.

x = 0, y = D input is going to circuit.

How to detect whether and gate has stuck at 0 on y?

Give x = 1, y = 1    if output is zero, then system has got a fault.

Why not x = 1, y = 0 not ??

Input does not change due to fault.

Why not x = 0, y = 1,

Fault will make it x = 0, y = 0

Output is same    xy = 00, xy = 01 are both zero output.

or : +    x + y

fault → stuck at 0 on y

How to detect ?

xy = 01

xor :

fault → stuck at 0 on y

How to detect ?

y should be 1, x can be any thing (x1)

Complementation fault

'and' gate    complement fault at y.

xy = 11, xy = 10

'or' gate → 01, 00

'xor' → 00, 01, 11, 10

Microprocessor Style Address Machine

MR: memory read $b \leftarrow [a]$

MW: memory write $[a] \leftarrow b$

a → memory address register

b → memory buffer register

c → pc

print (2*4) print (4*7) print (3*5)

Print (4*7)

code

[2*4]

```
ib<--4
s<--ib
ib<--7
t<--ib
ib<--s*t
print(ib)
```

$\boxed{[78] \leftarrow ib}$ →

```
b<--ib
ib<--78
a<--ib
MW
```

[3*5]

```
ib<--[78]
print(ib)
```

→

```
ib<--78
a<--ib
MR
ib<--b
```

```
0: ib<--12
1: print(ib)
2: ib<--10
3: ~~print(ib)~~ pc<--ib
4: ib<--34
5: print(ib)
6: ib<--47
7: print(ib)
8: ib<--82
9: Print(ib)
10: ib<--56
11: Print(ib)
12: ib<--91
13: print(ib)
```

Output → 12, 56, 91

'a' and 'b' can be used as normal register but 'c' cannot.

$\boxed{ib \leftarrow pc \text{ is valid}}$

c<--ib
pc<--ib } are same

Write boolean expression true for

$xy = 10$ only $\quad x \cdot y'$

$xy = 00$ and $xy = 11$

$xy + x'y'$ or $(x'y + xy')'$

$\{ xy / \cancel{+ x'y +} \}$

$xy + x'y' \rightarrow$ Sum of product

$(x+y').(x'+y) \rightarrow$ Product of sum

sum of product $\rightarrow$ 010 and 110

$x'yz' + xyz' = yz'$

Proof

$a + a'b = a+b$

$a + a'b = (a+a').(a+b)$

$= a+b$

$x.(x'+b) = x.x' + x.b$

$x + (x'.b) = (x+x').(x+b)$

010, 110, 111

$yz' + xyz$

$y(z' + xz)$

$yz' + xy$

- $a+b-c$ and $a-c+b$ are different

a: {p, q} $\quad$ b: {q, r} $\quad$ c: {p, r}

a + b: {p, q, r} $\quad$ a+b-c: {q}

a-c: {q} $\quad$ a-c+b: {q, r}

$a - b = a.b'$

c: true for $xy = 00$ and 10

$y'$

a:

$xy = 00$ and 01

$x'y' + x'y = x'$

b: true for $xy = 10$ and 01

$xy' + x'y$

Binary 5 → 101
9 → 1001

Let us assume there are 5 bits
then 5 → 00101
9 → 01001

2's complement for negative numbers (signed numbers)

Positive number is represented as it.
For negative number, bit by bit complementation is done and one is added.

+5 = 00101

−5 = 11011 = 11010 + 1

(11011)' = 00100 + 1 = 00101

+9 = 01001

−9 = 10110 + 1 = 10111

Add −5 and +9

−5 = 11011
+9 01001

100100

remove carry 00100 = +4

+9 and +9 added

01001
01001

10010

(10010)' + 1
= 01101 + 1
= 01110

Add −9 and +5

10111
00101

011100

11100 → is a negative no.

(11100)' + 1 = 00011 + 1 = 00100

−8 and −8

11000
11000

110000

01111 + 1

= 10000

+8 and +8

01000
01000

10000

01111 + 1

= 10000

Add −9 and −5

11011
10111

110010

negative number

(10010)' + 1 = 01101 + 1 = 01110

Every instruction is in 4 digits

- abcd means [ab] = [cd]
  5423 means in memory 54 put the content of 23.
- let memory 45 has 389. Memory 82 has 21345
  2745 now memory 27 has 389.
  9045 now memory 90 has 45 [not 389]
- '90' - is a working in a immediate mode.
- '0 to 89' → in normal mode

abcd means [ab] = [cd]
Following are exception -
- 90cd means [90] = cd
  not [cd]
- 93 memory will store [88] + [89]

- Write a program to stor 72 in memory 56
  9072, 5690

let [88] = 243
[89] = 432
[56] = 314
9356 → [93] = 675
it will always store [88] + [89]

- What will 8856 will do?
  [88] = [56] = 314
  [93] = 746

- 91 cd means print [cd]
- 97cd means [97] = cd and goto instruction cd
- 94 memory will store [88] - [89]
- 98cd means [97] = cd and goto instruction cd if [94] >= 0

- 156 is stored in memory 42?
  9078
  8890
  8990
  4293

- 240 in 59?
  9080
  8990
  8890
  8893
  5993
  is wrong!!

  9080
  8890
  8990
  5693
  8856
  5993

8893 → [88] = [88] + [89] is infinite

# Vector processor

$x = 12$, $y = 4$, $z = x+y$ (16)

In vector processor -

$p = [12, 34, 52, 9]$

$q = [4, 11, 5, 4]$

$r = p+q = [16, 45, 57, 13]$ in unit time

$t = 3:2:12 = [3, 5, 7, 9, 11]$

will not cross 12 (will be included if present)

$m = p.*q = [48, 374, 260, 36]$ in unit time

k is matrix

$$\begin{matrix} 2 & 4 & 6 & 7 \\ 8 & 3 & 5 & 9 \\ 7 & 2 & 5 & 4 \end{matrix}$$

$p = k[2,3] = 5$

element in 2nd row and 3rd column

$k[2,:] \rightarrow$ second row $= [8, 3, 5, 9]$

$k[:,3] \rightarrow \begin{matrix} 6 \\ 5 \\ 5 \end{matrix} = [6, 5, 5]^T$

- $A = B + C$

  B & C $\rightarrow$ 3 X 4 size

  for $i = 1:1:3$

  $A[i,:] = B[i,:] + C[i,:]$

  for $i = 1:1:4$

  $A[:,i] = B[:,i] + C[:,i]$

  $O(\overset{\min(m,n)}{m})$ $\rightarrow$ time complexity

  otherwise order $O(mn)$

$[p, q] = \text{size}(b)$

b(m,n) c(n,r) multiply

To get final matrix $O(m*n*r)$

a(i,j)

s = 0
for k = 1:1:n
    s = s + b(i,k) * c(k,j)

To get first row, we need $O(n*r)$

To get some element of final matrix (n) in ordinary.
In vector also n.

To get first row in ordinary → n*r
and in vector ----.

Matrix b

$$\begin{bmatrix} 6 & 2 & 3 \\ 1 & 7 & 5 \end{bmatrix}$$

Matrix c

$$\begin{matrix} 3 & 2 & 5 & 1 \\ 5 & 2 & 7 & 5 \\ 3 & 8 & 4 & 2 \end{matrix}$$

[6 2 3] .* [3 5 3] = [18 10 9]
[6 2 3] .* [2 2 8] = [12 4 24]
[5 7 4] = [30 14 12]
[1 5 2] = [6 10 6]

Above took r time

Make following matrix

$$\begin{matrix} 18 & 10 & 9 \\ 12 & 4 & 24 \\ 30 & 14 & 12 \\ 6 & 10 & 6 \end{matrix}$$

$O(n+r)$

Add columns n-1 times

$$\begin{bmatrix} 37 \\ 40 \\ 56 \\ 22 \end{bmatrix}$$

Time = $O((n-1)+r)$
Total time = $O(m*(n+r-1))$

# Tree Interconnection Scheme

add 4 7 5 9

(4 c1 7) in parallel add these numbers (9 c2 5)

(Like Binary Tree, reduce time to O(log n))

(11 c1 14)

25 → It takes 2 unit time.

There are many computers. Every computer has two inputs & one output. Every computer has two variables x & y.

input-left and input-right

4 is input-left of c1

7 is " right " "

x = input-left  y = input-right

z = x+y

output z.

→ following is program of addition of n numbers

Interconnection scheme

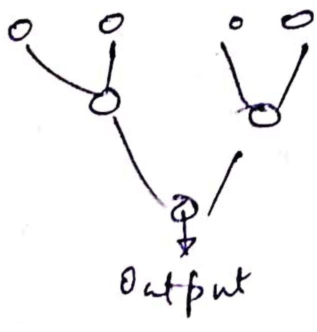
Computer K input-left is connected with computer 2K output.

computer K input-right is connected with computer 2K+1 output.

computer k sends its output to computer (k/2).

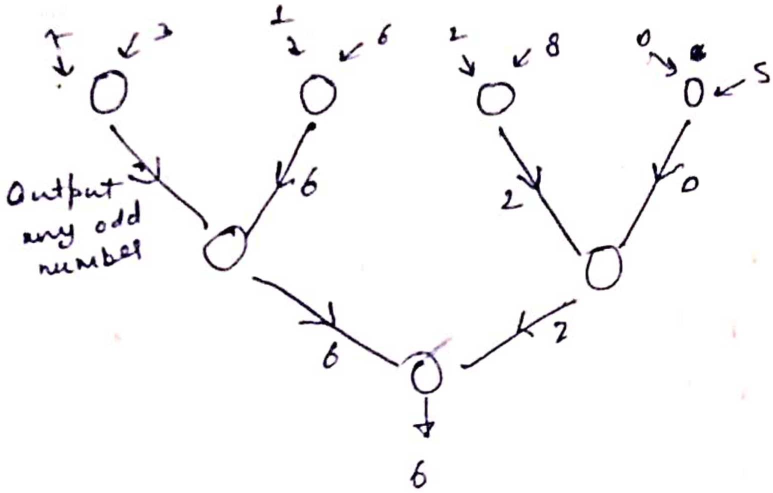
K is even → left

K is odd → right



• Write program to find first even numbers

7 3 1 6 2 8 0 5 → 6




$x$ = input - left , $y$ = input - right

if ( $x$ is even) , $z = x$

if ($x$ is odd) and if $y$ is even , $z = y$

else $z$ = any odd number

• If $t$ is even then $m = t$ else ~~m~~ = any odd.

⇧ can be reduced to

$m = t$

• Linear one way interconnection

Initial $y=0$

Let only one computer.

Following is done many times

$x$ = input

if $x < y$ $x \leftarrow\rightarrow y$

output $y$

$y = x$

first input from rights

input 6, 3, 5, 4, 2

First time

$x = 2$ $y = 0$

$x < 0$ false

output 0

$y = 2$

---

$x = 4$ , $y = 2$

output 2

$y = 4$

---

$x = 5$ $y = 4$

output 4

$y = 5$

---

$y = 5$ $x = 3$

$x = 5, y = 3$

output 3

$y = 5$

---

$y = 5$ $x = 6$

output 5

$y = 6$

```
# include <stdio.h>
  main()
{ int a,b;
   scanf ("%f", &a);
     b= . . . - - . . . ;
   printf ("%d %d" a,b);
}
```

input 12.4.

Method to store float numbers

12 = 1100

0.4 in float ~~for~~

0.0110 0110 0110 ---

0.4 x 2 = 0.8
0.8 x 2 = 1.6
0.6 x 2 = 1.2
0.2 x 2 = 0.4

0110

12.4 = 1100.0110011001100110 ---

32 bits to store float number

1st bit is sign bits

s <8 exponent bits> <23 mantisa bits>

s = 0 → positive

s = 1 for negative

1 abcd.... w * 2^n

exponent is 128 + n

mantisa is abcd ---- w

5.25 = 101.01 = 10.101 * 2^1     exponent is 2^1 is 128 + 1 = 129

.25 = 0.01 = 1.00 * 2^-2     exponent is 128 - 2 = 125

5.25 = 0<129><0101 00000.... >
(129 in 8 bits)

129 = 10000001

5.25 = 01000000 10101000 00000000

0.25     0<01111101><0 0 . . . . >

Always mantissa has 2 digits

1101L110 = ~~1.10 1110 * 2^~~ 1L01110 * 2^ 131

exponent = 131

~~{exponent}~~

0 <10000011> <1011110 . . . . . . . .

{ while writing mantissa
(1).01110
is ignored and 1011110

mantissa will have 2 digits

(1)1000001 0,100100 00000000 00000000

$11.001 * 2^2 = 1100.1$

-12.5

int a,

$\frac{525}{251 \times 256}$ $p = a/256$ $q = a \% 256$

```
{ int a, *k; float b; int p,q;
b = 5.25; K = &b; a = *k;
  a = a/256 ;  a = a/256;
  p = a/256 ;  q = a % 256
  printf ("%d %d", p,q);
}
```

p = 64
q = 168

b = 01000000 10101000 00000000 00000000

b = 01000000 00010000 — — —

- Multiplexer and Decoder(D)
  (selector) → method to design memory

→ S (12, 45, 67, 89, 56, 78 ; 3) → output 89
(0) (1) (2) (3)

; 8 → error

0 in place of 3 → 12

a = S (a, 45 ; k)

k = S (12, 45, 67, 89, 56, 78 ; 3)

K = 89

a = S (a, 45 ; K) What it does?

if K = 0 , ~~existing~~ a unchanged
K = 1 , a = 45

3125 ---> m31 <-- m25

S: selector takes multiple input single output
D: Decoder takes single input and output many numbers and others are zero.
Exactly one number is 1

D(3) --> 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, - - - -

D(0) --> 1, 0, 0, 0, - - ———

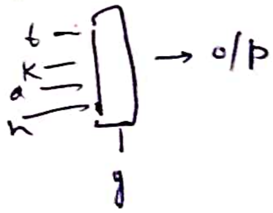
abcd does m[ab] <-- m[cd]

u = S (m[00], m[01], ... m[99] ; cd)

ki = D(ab)

mi = S (mi, u ; ki) for all i

S ( t, k, d, h ; y )



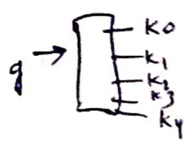

1247 is given

u is contents of memory 47

u = m[47]

D(12) --> k0 = 0, ... k11 = 0,
k12 = 1, k13 = 0 ... k99 = 0

D(k)




m[0] = S (m[0], m[47] ; 0) unchanged
upto m[11]

m[12] = S (m[12], m[47] ; 1) → changed
m[12] = m[47]

Following is exception -

abcd does $m[ab] = m[cd]$

exception 90cd does $[90] = cd$

$$\left( \begin{array}{l} m_i = S(m_i, u; k_i) \text{ for all } i \text{ and } i \neq 90 \\ m[90] = S(m[90], cd; k_{90}) \end{array} \right)$$

ab93 is $m[ab] = m[88] + m[89]$

ab94

$$\left( u = S(m[00], m[01], \ldots, m[92], m[88] + m[89], m[88] - m[89]; cd) \right)$$

$$m[94] = S(m[88] - m[89]; 0)$$

94cd will not modify $m[94]$

88cd will modify $m[94]$

97cd does $m[97] = cd$

98cd does $m[97] = cd$ if $m[94] >= 0$

$m[97] = S(m[97], cd; k_{97} + (m) [k_{98} \cdot (m[94] >= 0)])$ and

# Architecture of Parallel Computing

## Interconnection of computer (Linear one way scheme)

initial x0
y = input
if x > y   x <-> y
output x
x = y

output min(x,y)
x = max(x,y)

computer send output to its left computer

Let only one computer input 3,2,8,1,5

(x0, y) 3, 2, 8, 1, 5, infinite
(x0, y3) 2, 8, 1, 5
0 (x0, y3) 2, 8, 1, 5
---
(x3, y) 2, 8, 1, 5
(x3, y2) 8, 1, 5
2 (x3, y) 8, 1, 5
---
(x3, y) 8, 1, 5
(x3, y8) 1, 5
3 (x8, y8) 1, 5
3 (x8, y) 1, 5
---
(x8, y1), 5
1 (x8, y), 5
---
(x8, y5) ; i
5 (x8, y) i
---
(x8, yi)
8 (xi, y)
---

output - 0 2 3 1 5 8

$\boxed{n+1}$

array a1 .... an is given
for ( i=1, i<= n-1, i++)
if (a[i] > a[i+1])   a[i] <-> a[i+1]
Put biggest element to last
a[0] = 0 & a[n+1] -> infinite
above procedure is followed from i = 0 to n-1.

let two computers
[x0, y ] 0 ( x0, y) 3, 2, 8, 1, 5, i, i
[x0, y0] (x0, y3) , 2, 8, 1, 5, i, i
0 [x0, y] 0 (x3, y), 2, 8, 1, 5, i, i
---
[x0, y0] (x3, y2), 8, 1, 5, i, i
0 [x0, y] 2 (x3, y) 8, 1, 5, i, i
---
[x0, y2] (x3, y8), 1, 5, i, i
0 [x2, y] 3 (x8, y), 1, 5, i, i

[x2, y2] (x8, y1), 5, i, i
2 [x3, y] 1 (x8, y) 5, i, i

---

[x3, y1] (x8, y5), i, i
1 [x3, y] 5 (x8, y), i, i

---

[x3, y5] (x8, yi), i ...
3 [x5, y] 8 (xi, y), i ..

---

[x5, y8] (xi, yi) i ...
5 [x8, y] i (xi, y) i ...

---

[x8, yi] (xi, yi) i ...
8 [xi, y] i (xi, yi) i ...

---

[xi, yi] (xi, yi) i ...

Total time = $2(n-1)$

sorting n numbers,

n-1 computers is needed

n-1 are really needed when they are in reverse order and all distinct

How many computers are needed to sort?

1, 5, 2, 3, 4, 6, 9, 7, 8 only 1.

9, 1, 2, 3, 4, 5, 6, 7, 8 only 1

2, 3, 4, 5, 6, 7, 8, 9, 1 8

Linear interconnection: many computers connected in linear order

4 6 2 8 5 1 9 → 5 computers

• What is $a + a'$? Not always 1

Let $a = 1$ from time $t = 0$ to $t = 55$
$t = 55$ onwards $a$ is 0.
$a + a' = a + p$ $p = a'$

$p$ is 0 from time $t = 0$ to $t = 55$

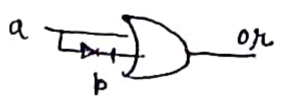
$t = 55$ onward $p$ is 1.

When delay of 3 units in not gate,

$p$ is 0, time $t = 0$ to $t = 58$

$p = 1$ $t = 58$ onwards

During $t = 55$ to $58$, $a$ and $p$ are both zero. Hence
$t = 55$ to $58$ $q = a + a'$ is 0.




not specifically this
because delay in or gate
because of transistors take time
to activate.

correct

~~If delay in or gate of 10 unit~~, then $t = 65$ to $68$ is zero.
~~and not gate also~~
~~$t = 0$ to $10$, output is 0.~~ → wrong
From $t = 0$ output is 1.

$a: 1 \to 0$ is problematic $a + a'$.

• $a \cdot b' + b$ is optimize as $b + a$. It is not correct.

Due to delay in not, and, or gates
not so important

Let $b = 0$ from $t = 0$ to $70$. Let $a = 1$ always.
delay not 7. delay or 10.
delay and 10.
when output is 0 during $t > 50$
→ never 0

~~{before $t \geq 70$ $a + b = 1$}~~

• $b = 1$ $t = 70 - 130$ let $b = 0$ $t > 130$

when output is 0 or gate delay 20
$a \cdot b' + b$
$p = b'$, $q = a \cdot b$, $r = q + b$

$t = 100$ onwards

upto 130 b is 1.

130 onwards $b = 0$ $t >= 130$

$p = 0$ upto 137 $\rightarrow$ $p = 1$ $t >= 137$

upto 147 $q$ is 0 $q = 1$ $t >= 147$

During $t = 130$ to 147 both $q$ and $p$ are zero

So, $t = 130 - 167 \rightarrow x$ is 0

---

- $p = x.z + y.z'$

  delay (and of $x.z$) $\rightarrow$ 20

  delay (and of $y.z'$) = 8

  delay (not) = 4

  delay (or) = 0

  let $t <= 50$, $z$ is 0 let $t >= 50$ $z$ is 1

  let $x$ and $y$ are 1 always

  During what time interval wrong output come $\rightarrow$ 62 - 70 - output is 0 (should be 1)

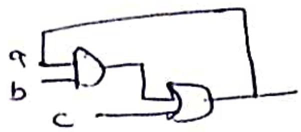
- $q = x.z + y.z' + x.y = x.z + y.z'$

  , During $q$ wrong output $\rightarrow$ never

  $p$ is optimized may give wrong output

- $K = a \cdot b + c$

  $a = a \cdot b + c$


b and c are inputs. Output is a.

$b = 0,\ c = 1 \qquad a = 1$

$b = 1,\ c = 1 \qquad a = 1$

$b = 0,\ c = 0 \qquad a = 0$

$b = 1,\ c = 0 \qquad a =$ retained / no change

$a = a \cdot b + c \cdot b' \rightarrow$ another expression ?? [product of sum form]

$b = 1$ retain

$b = 0,\ c = 1 \rightarrow 1$
$b = 0,\ c = 0 \rightarrow 0$ , $a = c$

$a \cdot b + c \cdot b' = a \cdot b + c \rightarrow$ $a = 0,\ b = 1,\ c = 1$ is difference between two expressions

- $a = a \cdot b + c \cdot b'$

  {~~[illegible]~~}

  $b = 0 \Rightarrow a = c \rightarrow$ general input

  $b = 1$ then retain

  controlled input

  $a = K(c, b)$

- $d = d \cdot b' + a \cdot b$ input a, b output d $d = L(a, d)$

  $b = 0$ , $d =$ retain

  $b = 1$ , $d = a$

b = 0, retain

b = 1, d = ~~[illegible]~~ a

b = 1, a is retained

b = 0 retain b = 1 retain

| | | | | |
|---|---|---|---|---|
| b = 0 | c = 0 | a = 0 | d = d | retain |
| b = 0 | c = 1 | a = 1 | d = d | retain |
| b = 1 | c = 0 | a = retained | d = a | since, a = retained (False Statement) hence d is retained |

d ≠ a previously, then d is not retained.

b = 1 c = 1 a = retained d = a

Initially, b = 0, c = 0, a = g, d = g

b = 0, c = 1, a = 1, d = g

b = 1 c = 1, a = retain = 1, d = 1

b = 1, c = 0, a = n = 1, d = 1

b = 0, c = 0, a = 0, d = n = 1

b = 0, c = 1, a = 1, d = n = 1

b = 0, c = 0, a = 0, d = 1

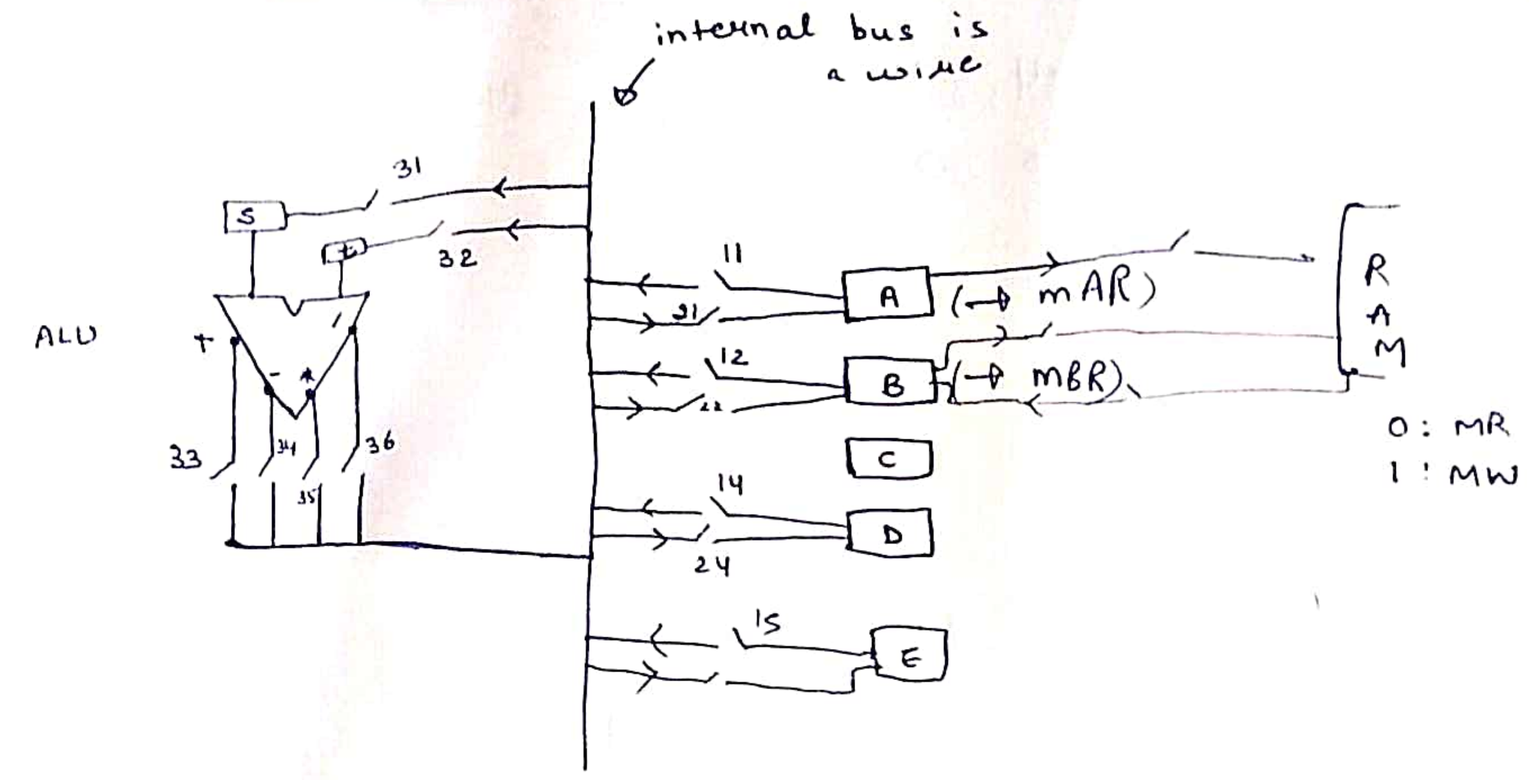
b = 1, c = 0, a = n = 0, d = 0

When output d changes -

When c changes but b unchange then no change in output

when b changes but c unchanged

b: 1 -> 0 no change, b: 0 -> 1 d = c




how we can perform ~~d~~ d = a+b this operation?

11, 12, 31, 33 → wrong

- 11 and 12 cannot be closed together (data conflict)

close 11, 31
open 11, 31
close 12, 32
open 12, 32
close 33, 24

close 11 : ib<--a
close 31 : s<--ib
open 11, 31
close 12 : ib <-- b
close 32 : t <-- ib
open 12, 32
close 33 : ib<--s+t
close 24 : d<--ib

swapping without using (c&d)

a <-> b    let a=79 b=83

close 11
close 31, 32    s=79 t=79
open 11, 31, 32

close 12, 21    a=83
open 21
close 34, 32    t=0
open ~~12~~ 32
open 32
close 22    b=79

BCD Numbers

binary 0101 + 0110 = 1011   carry = 0 Binary

0101 + 0110 = 0001   carry = 1   BCD

Binary numbers →

abcd + efgh = pqrs   carry y

$s = d \cdot h' + h' \cdot d$   $(c'g' + cg)$

$c_1 = d \cdot h$

$r = c_1' \cdot (c'g + g'c) + c_1 \cdot [(c'g + g'c)']$

add (cd, gh) condition of carry → $c \cdot g + ((+g) \cdot u$

let abcd + efgh is pqrs and carry y in binary
How to convert it in BCD ijkl and carry z.

1000 + 1001 = 0001   cy = 1 Binary
1000 + 1001 = 0111   cy = 1   BCD

---

0010 + 0011 = 0101   cy = 0 Binary

0010 + 0011 = 0101   cy = 0 BCD

---

1000 + 0101 = 1101   cy = 0 Binary
1000 + 0101 = 0011   cy = 1 BCD

---

1010 : A
1011 : B
1100 : C
1101 : D
1110 : E
1111 : F

5+7 = C , cy = 0   in binary   in BCD 2, cy = 1
8+9 = 1 , cy = 1   in binary   in BCD 0111 cy = 1
5+3 = 8, cy = 0   in binary   no change

$pqrs >= 10$

$p=1$, among $q$ and $r$ any should be one

$pqrs - 1010 = ijkl$

$l = s$

$K = r'$

$j = rq + r'q'$

$i = (carry)'$ | $i = 0$ under condition when $pqrs >= 10$ or $<= 2$

~~2-D interconnection One-way~~

→ 2-D one-way interconnection

p q r
s t u
v w x
y z m

→ Linear 2-way
a b c d e { a can send to b and b can send to a }

→ Linear 1-way : (we used it for sorting)

---

2-D one way interconnection scheme
(we will used this for matrix multiplication

Matrix A
1 2 3
4 5 6

Matrix B
7
8
9

Normally it will take 6 multiplications,

Now, Every computer has x and y. x for A and y for B

PAGE NO.:
DATE: / /

------ time t = 0 ------

(p ) 1, 2, 3
(q ) 4, 5, 6
7
8
9

------ time t = 1 ------ $a_{21} \cdot b_{11}$

(p ) 1, 2, 3
(q 4 7) 5, 6
8
9

$q = 7 * 4 = 28$

------ time t = 2 ------ $a_{22} \cdot b_{21}$, $a_{11} \cdot b_{11}$

~~p(1 7) 2, 3~~

(p 1 7) 2, 3
(q 5 8) 6
9

$p = 1 * 7$
$q = 28 + 5 * 8 = 68$

------ time t = 3 ------ $a_{12} \cdot b_{21}$, $a_{23} \cdot b_{31}$

(p 28) 3
(q 69)

$p = 7 + 2 \times 8 = 23$
$q = 68 + 6 * 9 = 122$

PAGE NO.:
DATE: / /

— time $t = 4$ — A — 9 (3, 63)

(p 3 9)
(q )

$p = 23 + 3 * 9 = 50$

Total time = 4

How much time will A: (23, 57) and B: (57, 1) take?

$\Rightarrow$ $57 + 23 - 1 = 79$

---

Example

Matrix A
7, 8, 9
1, 2, 3
4, 5, 6

Matrix B
7
8
9

at $t = 0$

(m ) 7, 8, 9
(p ) 1, 2, 3
(q ) 4, 5, 6

7
8
9

It will have to traverse through one more computer

& time taken = $3 + 3 - 1 = 5$

PAGE NO.:
DATE: / /

New Matrix A, and Matrix B

1, 2, 3

4 5
6 7
8 9

--- t=0

(p )(q ) 1, 2, 3
4 5
5 7
6 9

--- t=1

(p )(q 1 5) 2, 3
4 7
5 9
6

q = 1×5 = 5

--- t=2

(p 1 4) (q 2 7) 3
5 9
6

p = 1×4 = 4
q = 5 + 2×7 = 19

--- t=3

~~(p 2 5)~~ (p 2 5) (q 3 9)
6

p = 4 + 2×5 = 14
q = ~~5~~ 19 + 3×9 = 46

--- t=4

(p 3 6) (q )
p = 14 + 3*6 = 32

1:32 32:79

A : 1,32 B : 32,79

→ At time $t=1$ what is done?

- $a_{1,1} * b_{1,79}$

→ At $t=2$, what is done?

$a_{1,1} * b_{1,78}$

$a_{1,2} * b_{2,79}$

→ what is done at last $t$?

$a_{1,32} * b_{32,1}$

---

Now, Matrix A

2 3
4 5

Matrix B

6 7
8 9

- - - - - - - - - - $t=0$

(p )(q ) 2, 3
(r )(s ) 4, 5

6 7
8 9

- - - - - - - - - - $t=1$, $a_{21} \cdot b_{12}$

(p )(q ) 2, 3
(r )(s 47) 5

6 9
8

$s = 4 \times 7 = 28$

- - - - - - - - - - $t=2$ $a_{21} \cdot b_{11}$, $a_{22} \cdot b_{22}$, $a_{11} \cdot b_{12}$

(p )(q 27) 3
(r 46)(s 59)

8

$q = 2 * 7 = 14$

$r = 4 \times 6 = 24$

$s = 28 + 5 \times 9$

$=$

PAGE NO.:
DATE: / /

------ t = 3, $a_{11} \cdot b_{11}$
$a_{22} \cdot b_{21}$
$a_{12} \cdot b_{22}$

(p 26)(q 39)
(r 58)(s p)

p = 2 * 6
q = 3 * 9 + 14
r = 5 * 8 + 24

--- t = 4 ---, $a_{12} \cdot b_{21}$

(p 38)(q )
(r )(s )

p = 12 + 3×8

---

A = (50, 33), B = (33, 78)

when will C[10, 12] start

(50 − 10 + 1) + (78 − 12)

---

~~Time req also~~

for multiplying A(m, n)
B(n, r)

time req :- $m + r + n - 2$

# Pipeline Architecture

Instruction is done in 4 stages.

- Instruction Fetch (IF)
- Fetch Operand (FO)
- Compute (CP)
- Store Result (SR)

Let initial values of a, b, c, d are a = 23 b = 67 c = 82 d = 45

c = a+d  [0, 10) (IF)  [FO: 10, 20) (S = a = 23, T = d = 45)  [R = S+T : 20, 30) (= 68)  [c = R, 30, 40)

K = c+10  $\boxed{\text{A}=20 \text{ , } c = 82}$ $\boxed{\text{not } c = 68}$

[IF ; 10, 20)  [S = c = 82 T = 10 , 20, 30)  [30-40, R = S+T)  [K = R, 40, 50) (= 92)

| Time | 0 - 10 | 10 - 20 | 20 - 30 | 30 - 40 | 40 - 50 | 50 - 60 | 60 - 70 |
|---|---|---|---|---|---|---|---|
| c = a+d | IF | S = 23<br>T = 45 | R = 68 | c = R | | | |
| C = c+10 | | IF | S = 82<br>T = 10 | R = 92 | c = R | | |
| K = c+2 | | | IF | S = c = 82<br>T = 2 | R = 84 | K = R | |
| u = c+5 | | | | IF | S = c = 68<br>T = 5 | R = 73 | u = R |
| v = c+3 | | | | | IF | S = c = 92<br>T = 3 | R = 95<br>↓<br>V = R |
| j = 80 | | | | | { } | IF<br>{ } | j = 80 |
| K = b | | | | | | { } | IF<br>S = b<br>↓<br>K = 67 |

| | 70-80 | 80-90 | |
|---|---|---|---|
| print(K) | IF | S = K | print(K)<br>84 will be printed |

a = 23    b = 67    c = 82

d = 45    e = 26    f = 71

K: print(a)    IF    IF

print (b)    IF

print (c)    IF

goto K [pc=0]    IF    PC=0

print (d)    IF

print (e)

print(f)

o/p → 23 67 82 45 23 67 82 45 ...

- $0+0=0 \quad cy=0$
  $0+1=1 \quad cy=0$
  $1+1=1 \quad cy=1$

- In, $a+b$ condition for carry $= a\cdot b$
  Total time is $\Rightarrow$ 1 unit

- $ab+cd$ condition for carry $= ac+(a+c)\cdot bd$

Time 1 $\Rightarrow u=a\cdot c\ ;\ v=a+c\ ;\ w=b\cdot d$
Time 2 $\Rightarrow z=v\cdot w$
Time 3 $\Rightarrow u+z$

Total time is 3 unit.

- $gab+hcd$ condition for carry.

$$gh+(g+h)\cdot ac+(g+h)\cdot(a+c)\cdot bd.$$

$$\Rightarrow gh+(g+h)\cdot(ac+(a+c)\cdot bd)$$

Total time is 5 unit.

$gh;\ g+h;\ ac;\ (a+c);\ bd$
$(a+c)\cdot bd$
$u=ac+(a+c)\cdot bd)$
$v=(g+h)\cdot u$
$gh+v$

$\rightarrow$ 5 unit time

- Ripple Carry Adder -

add, abcdef + pqrstu
let carry of bcdef + qrstu = z.
then final carry = $a \cdot p + z \cdot (a+p)$

$t(n) = t(n-1) + 2$

carry of abcdef + pqrstu
⇒ let x is carry in def + stu ; y is carry in abc+pqr.
~~then~~ and $z = (a+p) \cdot (b+q) \cdot (c+r)$

let ABC are computers in parallel.

then we find - ~~x+y.z~~ $y + x \cdot z$

In this $T(n) = t(n/2) + 2$

carry of abc def + pqr stu ~~is true~~ has carry 1 when.

1) abc + pqr ⇒ cy = 1
2) def + stu ⇒ cy = 1
and
abc + pqr = 111

abcd is Binary Coded Decimal. 0 to 9

1010, 1011, 1100, 1101, 1110, 1111 are not possible

How to increment a BCD number?

input 0101 o/p 0110

input 1011 o/p Not possible

A BCD number abcd is incremented. Type condition for carry $a.d$ (when abcd = 9) $[a.b'c'.d]$

Binary number pqrs is a binary number. In this 1 is added. Condition for carry pqrs

uvhm is increment of BCD abcd. carry is t.

$t = a.d$

$m = d'$

$h = (d.c' + c.d')$ if abcd != 9

$+d').(dc'+c.d')$

$V = (cd \text{ xor } b).(a'+d')$ or $(cd \text{ xor } b)$

$u = (bcd \text{ xor } a) . (a'+d')$

$= bcd + d'.a$

Binary pqrs and 1 is added let xyzw is obtained and carry is g.

$g = p.q.r.s$

$w = s'$

$z = s.r' + s'.r = \text{xor}(s, r)$

$y = r.s.q' + (r.s)'q' = rs \text{ xor } q$

$x = q.r.s.p' + (q.r.s)'p = qrs \text{ xor } p$

Binary number abcd <= 7 ⟶ $a'$

BCD abcd <= 7 ⟶ $a'$ (same)

Binary number abcd >= 7 ⟶ $a + bcd$

BCD abcd >= 7 ⟶ $a + bcd$ (same)

{ Input $x, y$ o/P $z$

$z$ is 1 when $x = 1$ and $y = 0$ or $x = 0$ and $y = 1$

$z = x \text{ xor } y = x' y + x y'$

Input $p, q$ o/P $r$

$r = 1$ when $p = 1$ and $q = 0$

or $p = 0$ and $q = 1$

input $p, q = 11$ not possible

$r = p + q$

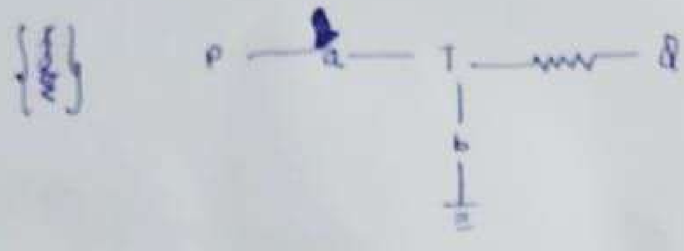
Take $p, q = 11$ as o/p $= 1$

$p' \cdot q + p \cdot q' + p \cdot q = (p' \cdot q')' = p + q$ }

- $a+b$

Between P and Q put switch a
" " " " " b

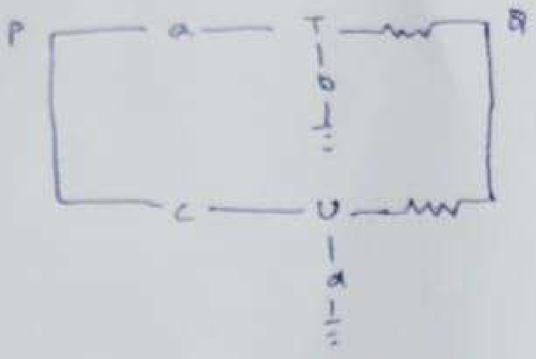
- $a.b'$





- $c.d' + a.b'$

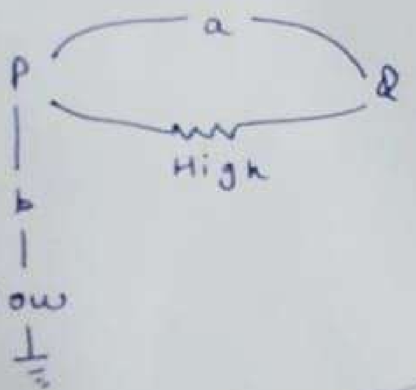
This is wrong why?

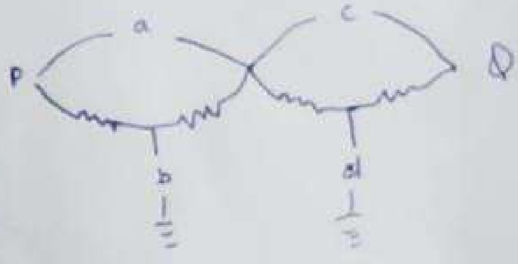



In this case, it will fail

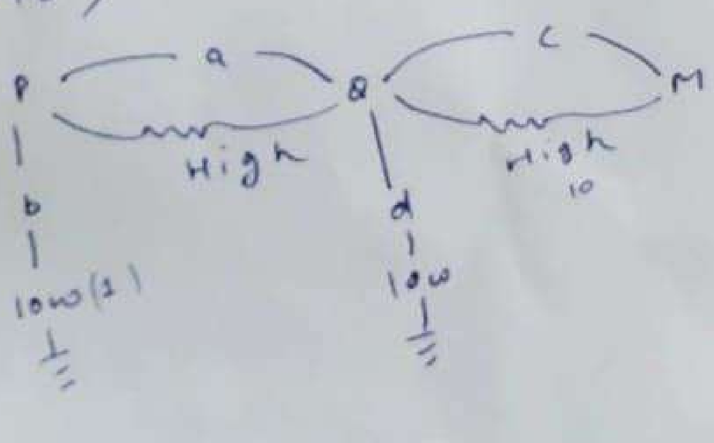
| | |
|---|---|
| ab 11 | cd 10 |
| ab 10 | cd 11 |

- $a+b'$




- $(c+d').(a+b')$




a = 1, b = 1 reach Q
c = 0, d = 0 PaQ10M
has more resistan

Pb1 earth has
less resistance

and $(x, y) = x.y$ $\quad x * y \quad$ y is 0 or 1.
or $(x, y) = x+y$ $\quad$ add x and y atleast one of x or y is 0

$a = c.b + a.b'$

$b=0$, $a$ = retain
$b=1$, $a = c$

There are two inputs & one of them is control input (b) and one is general input (c)

Edge Trigerred flip flop → o/p change only when control input change

b: control input 0, then o/p does not change even when input changes
when control input 1, then o/p is same as general input

| | | |
|---|---|---|
| c = 89 | b = 0 | o/p g: garbage |
| c = 67 | b = 0 | o/p same g |
| c = 67 | b = 1 | o/p 67 |
| c = 82 | b = 1 | o/p 82 |
| c = 82 | b = 0 | o/p 82 |
| c = 56 | b = 0 | o/p 82 |
| c = 92 | b = 0 | o/p 82 |

$a = c.b + a.b'$
$d = a.b' + d.b$

| | | | |
|---|---|---|---|
| c = 89 | b = 0 | a = g | d = g (same) |
| c = 67 | b = 0 | a = g (same) | d = g (same) |
| c = 67 | b = 1 | a = 67 | d = g (same) |
| c = 82 | b = 1 | a = 82 | d = g |
| c = 82 | b = 0 | a = 82 | d = 82 |
| c = 56 | b = 0 | a = 82 | d = 82 |
| c = 92 | b = 0 | a = 82 | d = 82 |
| c = 92 | b = 1 | a = 92 | d = 82 |

# Decoder and Encoder

## Decoder

$D(a, b)$ it has many outputs (binary)

$a$: enable binary

$b$: address general

$a = 0$ all outputs are zero

$a = 1$ then exactly one output is 1.

$D(1, 011) \rightarrow 00010000$

$D(0, \text{any}) \rightarrow 00000000$

$D(1, 110) \rightarrow 00000010$

$pqrstuvw = D3(a, bcd)$ or D8

Design D3 using D2's

$tuvw = D2(a \cdot b, cd)$

$pqrs = D2(a \cdot b', cd)$

$D3(a, bcd) = D2(a \cdot b, cd) \; D2(a \cdot b', cd)$

## Encoder

Has multiple input.

Atmost one input is 1.

output address of non-zero input

$E8(00100000) \rightarrow 1, 010$

$E8(00000000) \rightarrow 0, \text{any}$

$E8(pqrstuvw) \rightarrow a, bcd$

$E8(00100100) \rightarrow$ non-permitted input

$E4(0010) \rightarrow 1, 10$

$x, yz = E4(pqrs)$

$e, gh = E4(tuvw)$

$a = x + e$

$b = e$

$bcd = x \cdot (0yz) + e \cdot (1gh)$

$bcd = E7\,(pqrstuv)$ Atmost one input is 1.

Previously p was at location zero but now at location 1. output bcd

if o/p is 000 if all inputs are 0.

$E7(0010000) = 011$

$E3(000) = 00$

$E3(010) = 10$

E7 <u>using</u> E3

$E3\,(pqr)$

$\{E3(0,0,0)4\}$ $E3\,(stu)$

$(p,q,r)\,\{\,E3\,(stu)$

→ Pipeline Architecture : Instruction bypass -

let a = 12, b = 17, c = 13, d = 51

| | | |
|---|---|---|
| a = b + c | 0 − 40 | a = 30 |
| d = a + 9 | 10 − 50 | d = 21/39 |
| e = a + 15 | 20 − 60 | e = 27/45 |
| f = a + 19 | 30 − 70 | f = 31/49 |
| g = a + 20 | 40 − 80 | g = 50 |
| print(d) | 50 − 80 | |
| print(e) | 60 − 90 | |
| print(f) | 70 − 100 | |
| print(g) | 80 − 110 | |

Now the question is when a = 30 is successful.

We are sure a = 30 after t = 40, but before that we cannot be sure when it occurs.

Let some architecture have 1 instruction by pass
Output = 21 45 49 50

2 instruction by pass
output = 21 27 49 50

3 instruction by pass
" = 21 27 31 50

1 - instruction by - pass
if print (g) → garbage,
" (f) → 49
" (e) → 45
" (d) → 21

0101 ~~1010~~ 1011 ~~0110~~ 1001 + 1 10

# Negative number Multiplication

A number in 4 bits. −8 .. +7

+5 = 0101 , −5 = (0101)' + 1 = 1010+1 = 1011

The product of two numbers in 8 bits

(+5) * (+7) = +35 = 00100011

(−5) * (+7) = (1011) * (0111)

So our microprocessor will interpret as

11 × 7
= 77
= 01001101

But we do not want this behaviour

−5 * 7 = −35 what is expected answer

35 = 00100011

(−35) = (00100011)' + 1 = 11011100 + 1
= 11011101

## How to multiply negative numbers?

Subtract the Counterpart of negative number from first four bits of output

−5 * 7 = ~~100~~ (1011) * (0111)
= 01001101

Now substracting,

~~011~~ 01001101
0111 ~~0000~~
---
1101 1101

Last four bits will be as it is.

~~NOTE~~ ~~$x = 2^n$~~

PAGE NO.:
DATE: / /

Substracting binary

| | |
|---|---|
| $\begin{array}{r} 0100 \\ -\ 0111 \\ \hline 1101 \end{array}$ | similar to $\begin{array}{r} 435 \\ -\ 287 \\ \hline 148 \end{array}$ |

$\begin{array}{r} 287 \\ -\ 435 \\ \hline 852 \end{array}$

we just take the carry and forgot about it.

$+6 \longrightarrow 0110$ , $-6 \Rightarrow 1001 + 1$
$= 1010$

$-6 * 4 = (1010) * (0100) = 10 * 4$
$= 40$
$= 00101000$

$\begin{array}{r} 00101000 \\ 0100 \\ \hline 1101000 \end{array}$

NOTE $-x = 2^n - x$
where $x$ is in binary ~~representation~~,
and $n$ is no. of bits for representation

11101000
$= -[00010111 + 1] = -00011000$
$= -24$

~~1010~~ 1011

$\begin{array}{r}100\\ +1\\ \hline 0\end{array}$ ~~1000~~ ~~1010~~ ~~1010~~

$(-6)*(-5) = (1010)*(1011)$

$= 10 * 11$

$110$

$= 0110 1110$

~~0110 1110~~

~~1011~~

~~1010~~

~~1000~~

~~1010~~

~~0011~~

$$\begin{array}{r} 0110\ 1110 \\ -1011\phantom{\ 1110} \\ \hline 1011\ 1110 \\ -1010\phantom{\ 1110} \\ \hline 0001\ 1110 \end{array}$$

(Substracting −5 from first 4 bits)

(Substracting −6 from first 4 bits)

$= (30)$

abcd * pqrs if ~~aa~~ a.p = 1 then both are negative then we can treat them as positive but designing this type of circuitory with many if's is not easy.

Send Mail ~~not~~ related to

1) Why above ~~rules~~ multiplication logic is correct?

2) Logic for division?

Multiplication of two 2 digits number will be a 4 digit number.

Let a number in 2 base 10 digits.

0 ... 99 -50 .. +49

How a negative number is represented → by adding 100.

What is -35 → -35+100 = 65

40 * -20 = 40 * 80 = 3200

3200
-40
9200
-800

0 to 49 are +ve
50 to 99 are -ve

How to multiply 40 * 70 = 2800
→ let 0 to 80 are positive → (0 to 8000 are positive)
then ans 2800 is correct

→ Now, if 0 to 60 are positive
2800
-40
8800 = -1200

→ this means in 4 digits (0 to 6000 are positive)

(0 to 3000 are positive)

→ Now if let 0 to 30 are positive

2800
40
8800
7000
1800 →

checking correctness
40 → -60 (>30)
70 (>30) → -30
(-60)(-30) = 1800

PAGE NO.:
DATE: / /

So, Range need not to be 50-50

(In binary we take range (for positive and negative) to be 50-50 because we determine sign by MSB, but that is not necessary)

(we can divide range any way we want and answer will depend upon that :)

→ There are three possible ~~ans~~ answers for 40 * 70 depending upon how range is divided
→ both positive, one negative, both negative

How (1800) % 40 is done for different range?

- Notation : [a, b, c, d ; T] means

  $ax + by + cz + dw >= T$

  a, b, c, d are numbers and x, y, z, w are boolean variables 0/1

$[5, 7, 9, 20; 25] \to 5x + 7y + 9z + 20w \geq 25$

$w.(x + y + z)$

$[5, 7, 9, 15; 17] \to w(x+y+z) + w'xyz = w(x+y+z) + xyz$

What type of boolean expressions can be represented by it

Let [12, 56, 34, 78, 53; 143] be P (x, y, z, w, u)

Let [12, 56, 34, 78, 53; 113] be Q (x, y, z, w, u)

$P \to Q$

If P is true, then Q is true

Only following type can be represented

$v.Q + P$ where $P \to Q$ [P and Q are expression without v]

$v.Q + v'.P$

$W.(z + xy) + z(x+y)$

Design for $z(x+y)$ and $z+xy$ and here $z(x+y) \to z+xy$

[2, 4, 10; 11] [2, 4, 10; 6]

[5, 7, 20; 22] [5, 7, 20; 12]

$w.(z+xy) + z(x+y) = [5, 7, 20, 10; 22]$

22 - 12

$z + x'y$

[-2, 9, 12; 8]

# SR and JK flip flop

let input a, b output are p, q

$p = a + q'$
$q = b + p'$

ab = 01   pq = 01
ab = 10   pq = 10
ab = 11   pq = 11   [ input 11 never ]
ab = 00   pq = retains both

## JK flip flop

Input u, v
output is pq

$a = u.q$
$b = \cancel{(p)} v.p$
Then SR flip flop

uv = 01 , initial pq = 01
a = 0
b = 0
then retain as 01

uv = 01 intial pq = 10
a = 0
b = 1 then 01

uv = 01 then pq = 01 by either way

uv = 00 pq = retain

uv = 10 is similar to uv = 01

uv = 11 initial pq = 01
ab = $\phi 0$ , $\phi 0$
pq is complemented

add (0,0) = no carry (n)
add (1,1) = carry (c)
add (0,1) = n

Let a and b be digits.
add (0a, 0b) = n (N) [0,0]
add (1a, 1b) = c (C) [1,1]
add (1a, 0b) = undecided (U) [0,1]

[1,0] will be used later

~~{add (p, q)}~~

add (abcd, efgh) How to determine carry status.
abcdefghij are 0/1.
Let xyzw be N, C or U.
Let add (ai, ej) = x
add (bi, fj) = y
add (ci, gj) = z
add (di, hj) = w
x/y/z/w = (x/y)/(z/w)
Let p, q, r, s be N, C, U

p/q = C if p is C
p/q = N if p is N
p/q = q if p is U

How to find of add (abcd, efgh)
add (abi, efj) = p
add (cdi, ghj) = q
Now p/q

carry status
add (ai, bj) = [a·b, a+b]

- ABCD--- YZ are states.

0, 1, 2, 3 are inputs

let initial state is A.

let input string is 02013

A0B2E0G1G3G

Automata (A0B) (A3F) (B1C)(B3E) (B2E) (C1D) (C3F) (E0B)

(E0G) (D3A) (E1Y)

let input 0231

A0B2E3E1Y

let input 013

A0B1C3F → not correct

A0B1D (not C) 3E

- (A0B) (B1C) (C1B) (B3Y)(C3W)

0 1 3

A0B1(toggle BC) 3 (toggle YW)

wrong

A0B1(toggle BC) 3 (Y or W uncertain)

If B3Y is not true then after 3 (B or W uncertain)

- (P0Q) (P1S) (Q3R) (T3U) (S3U)

(Q2T)

Input → 03 initial P

P0Q3 (not R)

P0Q3U

(01/10) [S/T] 3U

- ~~(A0B)~~ (B1C) (B0E) ~~(B3G) (C2D) (E0H)~~
- ~~(A0B) (B1C) (B3G) (C2D) (D3H) (H3K)~~

- (A0B)(B1C)(B3G) (G2H)(H2J) ( J2H) (B2K)

012

A0B1C2C → wrong

1: 01 , 2: 10

012 is {eg} 0102 or 0132

A0B1C0C2C or A0B1C3C2E

{~~A0B3G2(JH toggle)~~}

- x + y

{how - xy = 01 is given}

xy = 01 to 10 how to transit?

01 to 10 should be via 11.

- $x \cdot y + z \cdot y' + x \cdot z$ optimize it

$= xy + z \cdot y'$

{~~xyz~~ + ~~xy'z~~}

- Difference???

• $S(a, b, c, d; p, q)$

$p, q = 00$ o/p : a

$p, q = 01$

$p, q = 10$

$p, q = 11$

$S(a, b, c, d; y, y')$ will output what?

{a and d will never be outputted} → not always correct

when delay then 'a' 'd' can be output

$y$ 0 → 1    $y$ 1 → 0

a and d can be outputted for short time.

$z = S(a, z, z, d; p, q)$

$p=0$ $q=1$ retain

$a = 56$ $d = 67$ $pq = 11$, $z = 67$

$a = 56$ $d = 67$ $pq = 01$, $z = 67$

$a = 56$ $d = 89$ $pq = 01$, $z = 67$

$a = 56$ $d = 89$ $pq = 00$, $z = 56$

$a = 56$ $d = 89$ $pq = 01$, $z = 56$

$z = S(a, z, z, d; y, y')$

$a = 56$ $d = 67$ $y = 1$, $z = g$ stable

$a = 56$ $d = 67$ $y = 0$, $z = 56$ (due to delay) or same g (no delay)

$a = 74$ $d = 67$ $y = 0$ $z = 56$

$a = 74$ $d = 67$ $y = 1$ $z = 67$ → stable

delay of S is 10 sec → shift output
delay of not is 2sec → duration delay

a = 56  d = 67  y = 1  o/p g = 81

At t = 50  y : 0

during t = 50 - 52 both y and y' = 0

during 60 - 62 z is 56.

t = 52 - 60  y = 0  y' = 1  hence  z = retain (t - 10)

during 62 onwards → z is 81 again

| | |
|---|---|
| 62 - 70 | z = 81 |
| 70 - 72 | z = 56 |
| 72 - 80 | z = 81 |

- u = S (a, b ; w)

w = 0  u = a  w = 1  u = b

Delay in S is 10

a = 34  b = 67

w : 0 -> 1, at t = 50

at t = 60, u changes from 34 to 67

a : 34 -> 55 at t = 47

at t = 50 w : 0 -> 1

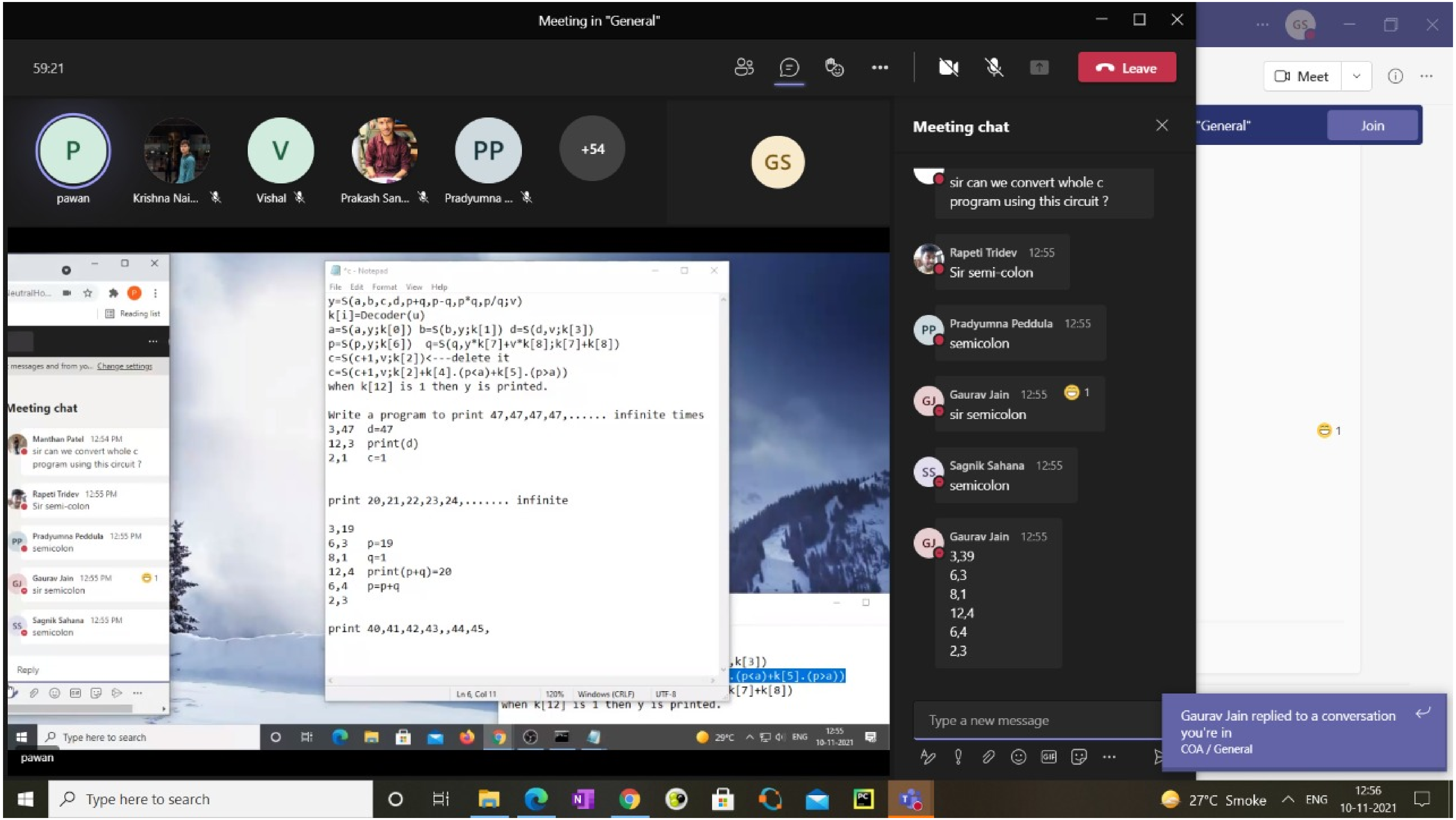
- v = S (d, v ; k)

d = 81 -> 25 at t = 47

K = 0

what is o/p v ?  0 - 57, v = 81

K 0 → 1 at t = 50  t > 57, v is 25



Meeting in "General"
59:21
Leave
pawan
Krishna Nai...
Vishal
Prakash San...
Pradyumna ...
+54
GS
Meeting chat
sir can we convert whole c program using this circuit ?
Rapeti Tridev 12:55
Sir semi-colon
Pradyumna Peddula 12:55
semicolon
Gaurav Jain 12:55
sir semicolon
Sagnik Sahana 12:55
semicolon
Gaurav Jain 12:55
3,39
6,3
8,1
12,4
6,4
2,3
Type a new message
Gaurav Jain replied to a conversation you're in
COA / General
Meet
"General"
Join
*c - Notepad
File Edit Format View Help
y=S(a,b,c,d,p+q,p-q,p*q,p/q;v)
k[i]=Decoder(u)
a=S(a,y;k[0]) b=S(b,y;k[1]) d=S(d,v;k[3])
p=S(p,y;k[6]) q=S(q,y*k[7]+v*k[8];k[7]+k[8])
c=S(c+1,v;k[2])<---delete it
c=S(c+1,v;k[2]+k[4].(p<a)+k[5].(p>a))
when k[12] is 1 then y is printed.
Write a program to print 47,47,47,47,...... infinite times
3,47 d=47
12,3 print(d)
2,1 c=1
print 20,21,22,23,24,....... infinite
3,19
6,3 p=19
8,1 q=1
12,4 print(p+q)=20
6,4 p=p+q
2,3
print 40,41,42,43,,44,45,
Ln 6, Col 11
120%
Windows (CRLF)
UTF-8
Meeting chat
Manthan Patel 12:54 PM
sir can we convert whole c program using this circuit ?
Rapeti Tridev 12:55 PM
Sir semi-colon
Pradyumna Peddula 12:55 PM
semicolon
Gaurav Jain 12:55 PM
sir semicolon
Sagnik Sahana 12:55 PM
semicolon
Reply
Type here to search
27°C Smoke
ENG
12:56
10-11-2021

- Let $a, b, c, d, e$ be digits. Base 10

$5a + 7b \to$ carry = 1

$3a + 4b \to$ carry = 0

$3a + 6b \to$ $a+b >= 10$ then carry 1 (undecided)
otherwise carry 0

~~Sum~~ if $a+b = 9$ then undecided

when two numbers are added and only their first digits are known then 3 possibilites

carry 0, carry 1, carry undecided

- Let 3 numbers are added.

Only first digits are known.

what are possible carries?

Sum of first digits let k

k: 0 ... 7 carry 0

k = 8, 9 carry 0/1

k: 10 ... 17 carry 1

k: 18/19 carry 1/2

K: 20 .. 27 carry 2

Access 8

Numbers are -8 to +7

Representation in 4 bits abcd

Every no. is represented by adding 8.

+3 is 11 -> 1011

-5 is 3 -> 0011

0101 -> 5-8 = -3

+7 is 15 -> 1111

-7 is 1 -> 0001

-8 is 0 -> 0000

How to add? Normal addition in 5 bits MSB is carry.

add +3 and -5

1011 0011

0011 Normal add

01110

0110 - 1000 = -2

carry is transferred to MSB

add +3 and +7

1011

1111

11010

+3 and +2

1011

1010

10101

1101 - 1000 - 0101

- Normal addition. let pqrst. Ans: prst

Multiply +5 and +7

+5 = 1101 = 13

+7 = 1111 = 15

expect +35

normal multiplication = 195

1010 0011 is expected

normal: 1100 0011

Multibly +5 and -3

output in access 128

+5 is 1101

-3 is 0101

normal multi = 65
= 0100 0001

-15 + 128 = 113 = 0111 0001

Multiplication Rule?

abcd is sign megnitude

+ = 0
- = 1

a: is sign    bcd: magnitude

+5 is 0101

-5 is 1101

+7 is 0111

-7 is 1111

pqrs is 2's complent

uvwx is access 8

express them in terms of each other.