

Subject :

Ch - 1 - 1  
Date

90, 11, 19

Subject

OS -  
جی اے اس

( Register Transfer Level Description ) RTL

ar - | داد میں اسی طرز تھا .  
Cpu -  
io -

CPU clock, clear, t

rd CPU

( dubia ) Pipeline

کامپیوٹر کی پروگرام

Microprogrammed

I/O

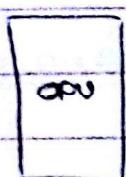
،  
Gibson .

2. Computer System Architecture by M. Mano

3. Computer Organization and Design by D. Patterson & J. Hennessy

4. Computer System Organization by W. Stallings

Op -  
Op -  
Op -



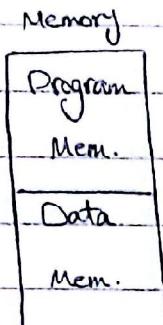
CPU : Central Processing Unit ( Cpu )

CPU

s.a.m

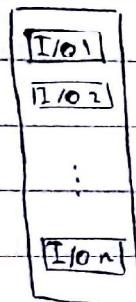
میکرو پردازنده CPU

1. Fetch  $\rightarrow$  عالی ترین سطح
  2. Decode  $\rightarrow$  میانی سطح
  3. Execute  $\rightarrow$  پایین ترین سطح



Instruction Memory	( $\sim 10^4$ words)
Code Memory	( $\sim 10^4$ words)

Program Memory is Shared



$(\text{User})$ $\cup$ $\vdash$ $\vdash$	$\vdash$ $\vdash$ $\vdash$
---	----------------------------------

Up for a BUS tour to see the world.

Data BUS ( $2^k$  bits)

BUS - Address BUS (CS, OE)

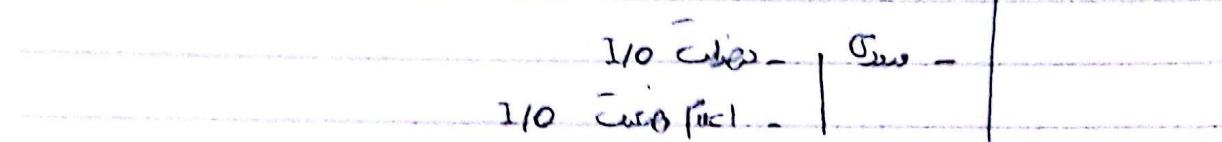
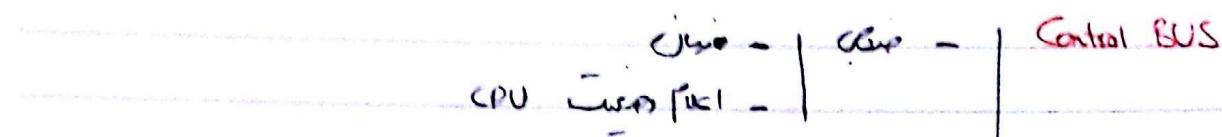
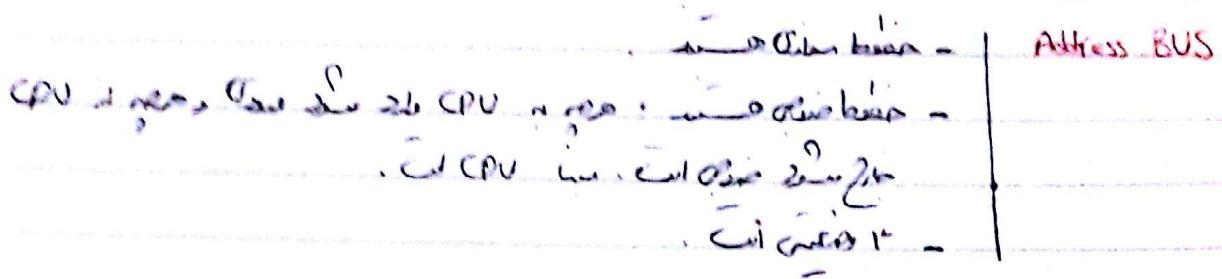
Control BUS (CPU)

مُعَادِلٌ ( Parallel ) ~~وَسِيمَيْكِي~~ - ۱. داٹا بیس

( Bidirectional ) ~~incomplete~~

• (3-State) Culcias r -

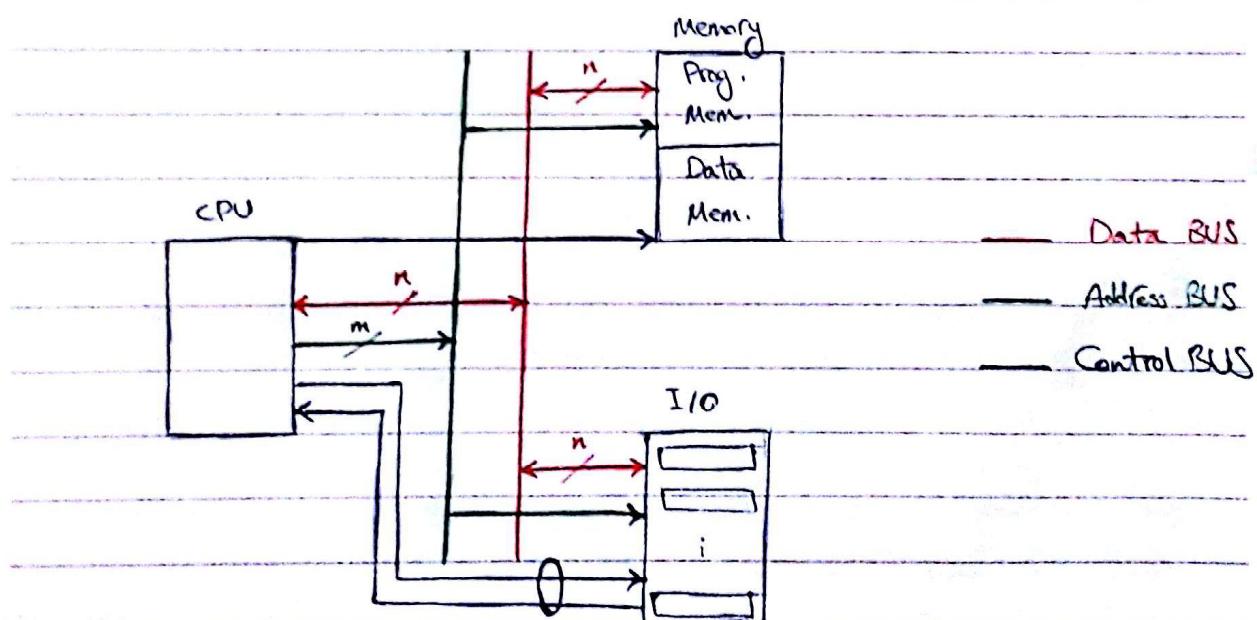
s.a.m



(الخطوة الأولى - خطوة الثانية) (الخطوة الثالثة) -  
ـ مقدمة إلى الحاسوب (الخطوة الرابعة) -  
ـ مقدمة إلى الذاكرة (الخطوة الخامسة) -  
ـ مقدمة إلى المدخلات وال 输出 (الخطوة السادسة) -

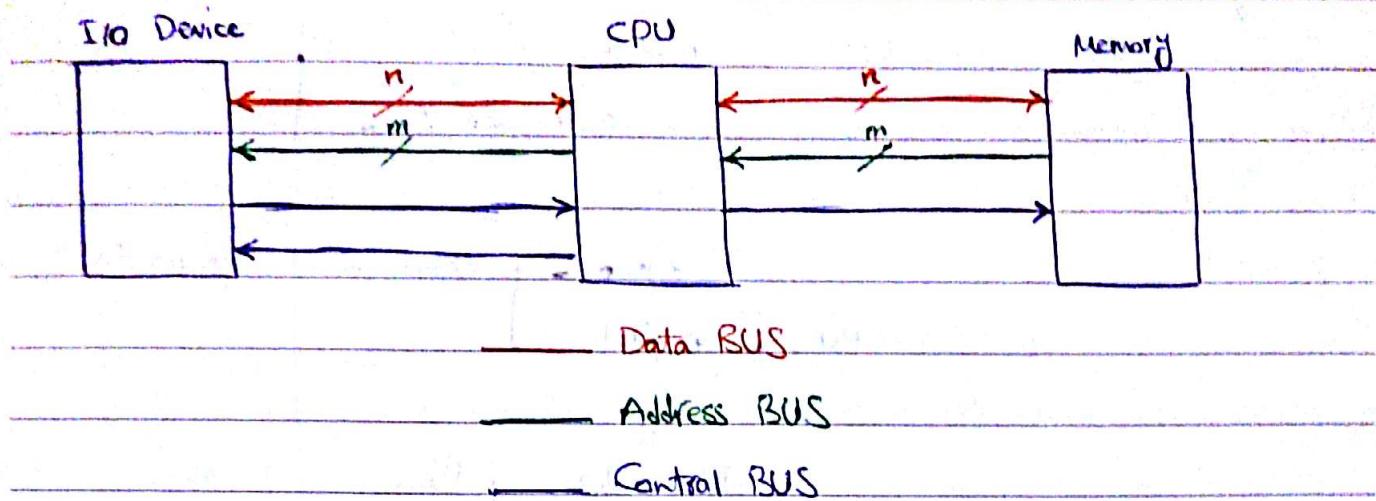
الخطوة السابعة

### 1. Von Neumann Architecture



s.a.m

## 2. Harvard Architecture



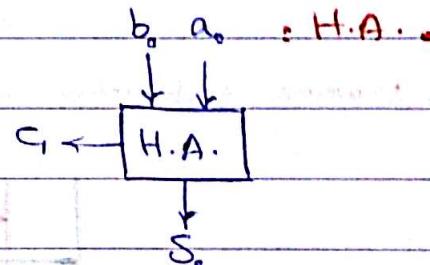
(Super Harvard Architecture) Share  $r_i, s_b, s_p, p_o, c$  to  $I/O$   $\rightarrow$

90, 11, 11  
10, 10, 10  $\rightarrow$   $c$

0 0 1 P

0 1 0 1  
— — — —  
1 0 1 0

$\rightarrow$  C (Carry)



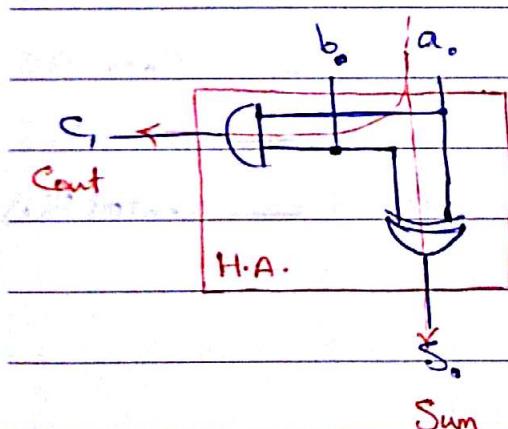
$a, b, | c, S$

0	0	0	0
---	---	---	---

0	1	0	1
---	---	---	---

1	0	0	1
---	---	---	---

1	1	1	0
---	---	---	---



s.a.m

$$T_{\text{sum}} = T_{\text{XOR 2}}$$

$$T_{\text{out}} = T_{\text{AND 2}}$$



$b_k \quad a_k$   
 $\downarrow \quad \downarrow$   
 $c_{k+1} \leftarrow \boxed{\text{F.A.}} \leftarrow c_k$   
 $\downarrow$   
 $s_k$

: F.A.

$a_k$	$b_k$	$c_k$	$c_{k+1}$	$s_k$
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	1

$c_k$	00	01	11	10
0	0	0	1	0
1	0	1	1	1

$$= c_{k+1}$$

$$= b_k c_k + a_k b_k + a_k c_k$$

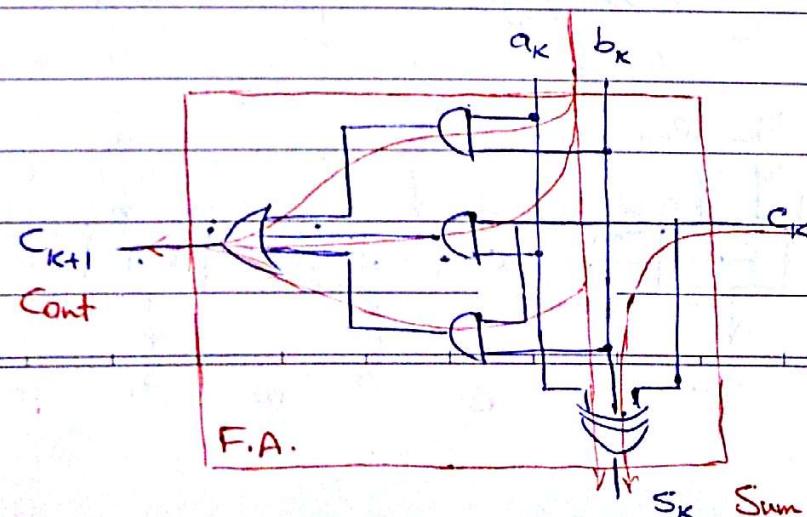
$c_k$	00	01	11	10
0	0	1	0	1
1	0	1	1	0

$$= s_k$$

$$= a_k \oplus b_k \oplus c_k$$

XNOR  $\equiv$  XOR  $\oplus$  Cont  $\oplus$  C<sub>k+1</sub>, Sum  $\oplus$  s<sub>k</sub>  $\oplus$  c<sub>k</sub>

s.a.m



$$T_{\text{sum}} = T_{\text{XOR } 3}$$

$$T_{\text{cont}} = T_{\text{AND } 2} + T_{\text{OR } 3}$$

$c_k$	$a_k b_k$	$\infty$	$01$	$11$	$10$	$11$
0	0	0	1	0		
1	0	1	1	1	1	

$$\begin{aligned} c_{k+1} &= a_k \cdot b_k + \bar{a}_k \cdot b_k \cdot c_k + a_k \cdot \bar{b}_k \cdot c_k \\ &= a_k \cdot b_k + (\bar{a}_k \cdot b_k + a_k \cdot \bar{b}_k) \cdot c_k \end{aligned}$$

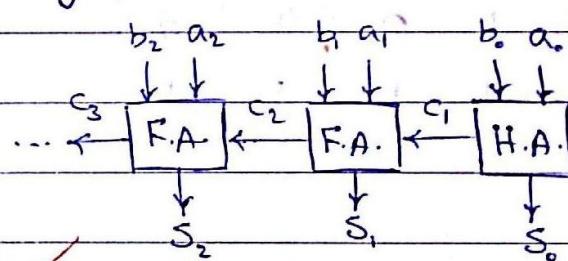
$$c_{k+1} = a_k \cdot b_k + (a_k \oplus b_k) \cdot c_k$$

: n-bit Carry

### ① Ripple Carry Adder (RCA)

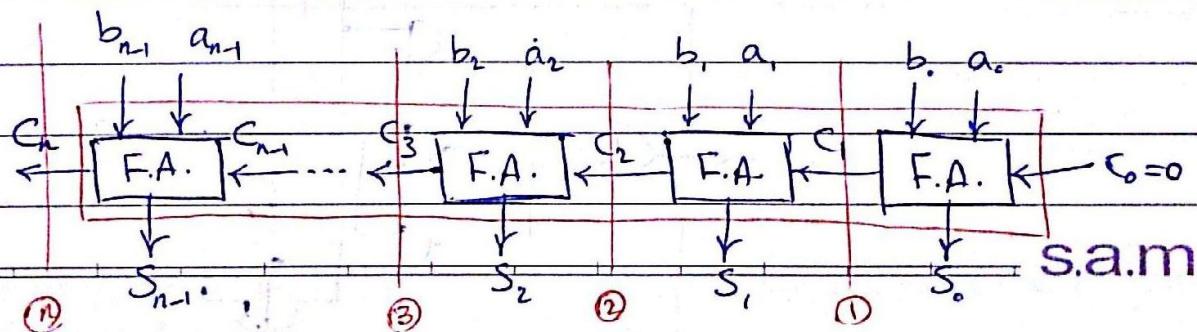
$$\begin{array}{r}
 01111111 \\
 100101110 \\
 + 00101110 \\
 \hline
 11000100
 \end{array}$$

### Carry Propagate Adder (CPA)



RCA

Sum bits,  $b_1, b_2, b_3, b_4$ , F.A. - H.A.  $S_0, S_1, S_2, S_3, S_4$



$A = (a_{n-1}, \dots, a_0)$  (n-bit)

$B = (b_{n-1}, \dots, b_0)$  (n-bit)

$$S = A + B$$

و ...  $C_3$  و  $C_2$  و ...  $C_1$  و  $C_0$  .  $C_i$  معرفه شده است که  $C_i$  نتیجه از  $C_{i-1}$  است .  
 (ریپل کاری) Ripple Carry میگویند که  $C_i$  برابر با  $C_{i-1}$  است .

$$T_{\text{sum}} = T_{XOR3}, T_{\text{cout}} = T_{AND2} + T_{OR3} \rightarrow \text{F.A.}$$

$$T_{n\text{-bit RCA}} = (n-1) \times T_{\text{cout}} + \underbrace{\text{Max}(T_{\text{cout}}, T_{\text{sum}})}_{\text{ناتج از } (n-1) \text{ دفعه}} + T_{\text{cout}}$$

!  $T_q$  (Max) ناتج از  $T_{\text{cout}}$  و  $T_{\text{sum}}$  است .

$$T_{n\text{-bit RCA}} = (n-1) \times (T_{AND2} + T_{OR3}) + \text{Max}(T_{AND2} + T_{OR3}, T_{XOR3})$$

با خواسته برای  $(T_q)$  ناتج از  $T_{\text{cout}}$  است .

$$T_{n\text{-bit RCA}} = (n-1)(2T_q) + 2T_q = n \times 2T_q \rightarrow \boxed{2nT_q}$$

و ...  $C_3$  و  $C_2$  و ...  $C_1$  و  $C_0$  .  $C_i$  معرفه شده است که  $C_i$  نتیجه از  $C_{i-1}$  است .  
 (F.A.) F.A. میگویند که  $C_i$  برابر با  $C_{i-1}$  است .

!  $T_q$  ناتج از  $T_{\text{cout}}$  است .  $T_{\text{cout}}$  ناتج از  $T_{\text{sum}}$  است .  $T_{\text{sum}}$  ناتج از  $T_{XOR3}$  است .

لذیعیتی مکانیزم RCA نہیں ~~\*~~

## ② Carry Look Ahead (CLA)



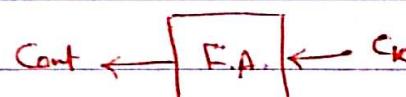
برای این دو مرحله میان زنار سه کم را می بینیم و این F.A. می باشد که می بینیم که این سه کم را ایجاد کردن خواهد شد.

معنی: طبق ماده می باشد  
معنی: ازین بدل طبقه (ترین) خواهد بود که حاصل!

حاصل بود که حاصل می باشد!

$$\text{Cont}_{\text{F.A.}} = \underbrace{a_k \cdot b_k}_{=g_k} + \underbrace{(a_k \oplus b_k) c_k}_{=p_k}$$

if  $g_k = 1 \rightarrow \text{Cont} = 1$



if  $p_k = 1 \rightarrow \text{Cont} = c_k$

$g_k$  : Carry Generator

$p_k$  : Carry Propagator

$$C_1 = g_0 + P_0 C_0$$

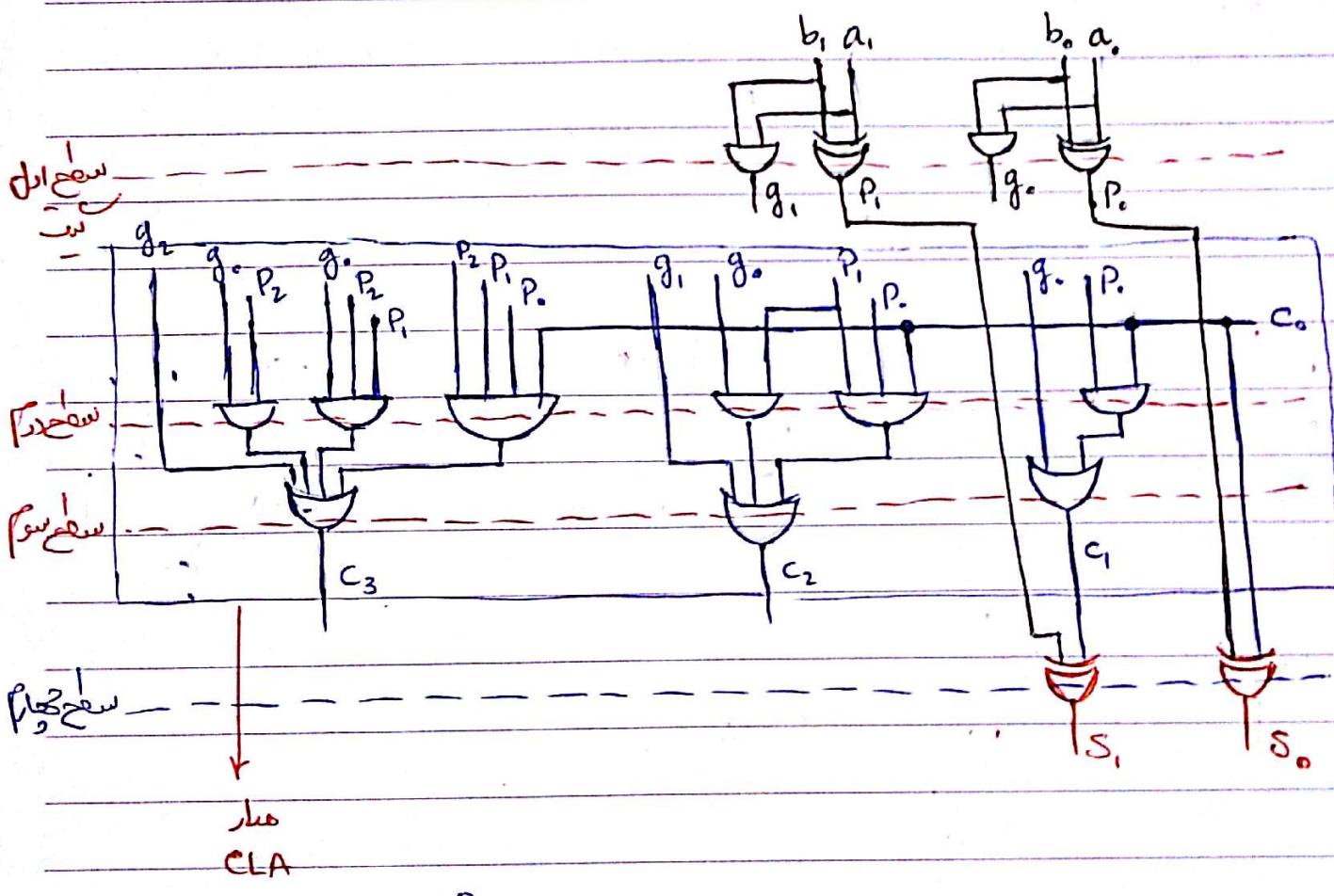
$$C_2 = g_1 + P_1 C_1 = g_1 + P_1 g_0 + P_1 P_0 C_0$$

$$C_3 = g_2 + P_2 C_2 = g_2 + P_2 g_1 + P_2 P_1 g_0 + P_2 P_1 P_0 C_0$$

$$C_4 = g_3 + P_3 C_3 = g_3 + P_3 g_2 + P_3 P_2 g_1 + P_3 P_2 P_1 g_0 + P_3 P_2 P_1 P_0 C_0$$

:

$$C_8 = g_7 + P_7 g_6 + P_7 P_6 g_5 + P_7 P_6 P_5 g_4 + \dots + P_7 \dots P_0 C_0$$

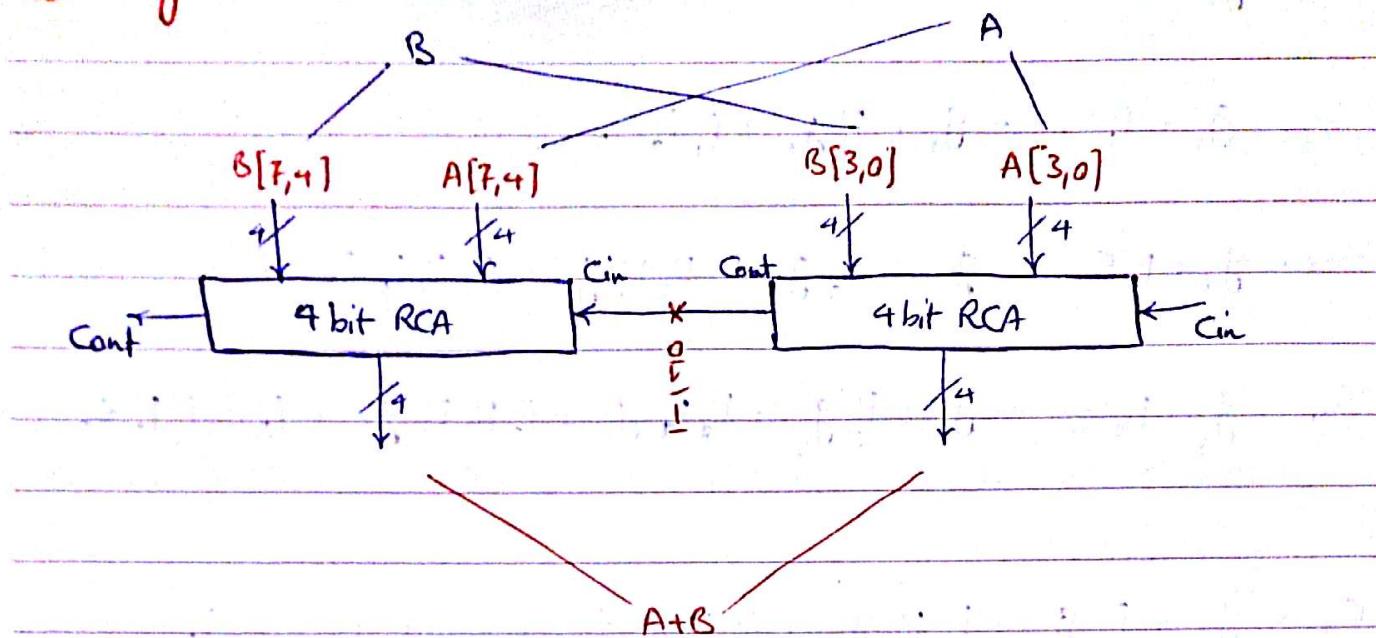


$$= P_k$$

$$\text{Sum} = S_k = (a_k \oplus b_k) \oplus c_k$$

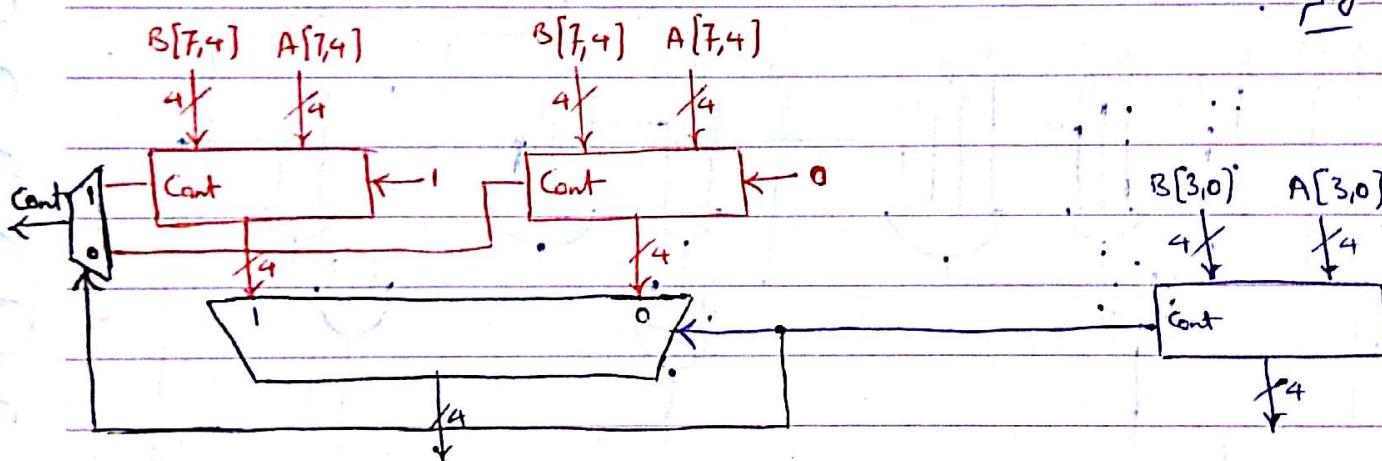
s.a.m

### ③ Carry Select Adder



$$\frac{T}{T_{RCA}} = 16 T_C$$

RCA مادیں اور مادیں، Cont مادیں کہنے، جو ایکسپریس اور ٹی اس سے اور بھرپور اور مادیں کہنے، هر سچے Cont اور بھرپور اور مادیں کہنے، میں از اولن درستہ درائیں اور مادیں کہنے!



$$T = 8T_g + T_{\mu\omega} = 10T_g$$

ANSWER

color,

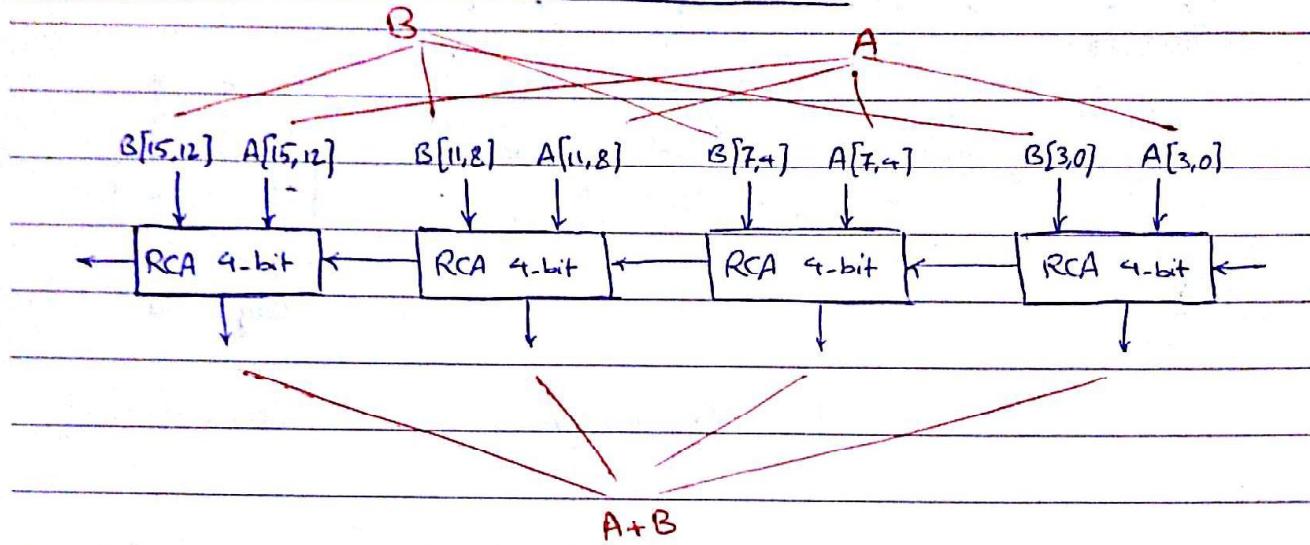
2T<sub>9</sub> 515 σ

امان

90, 11, 10

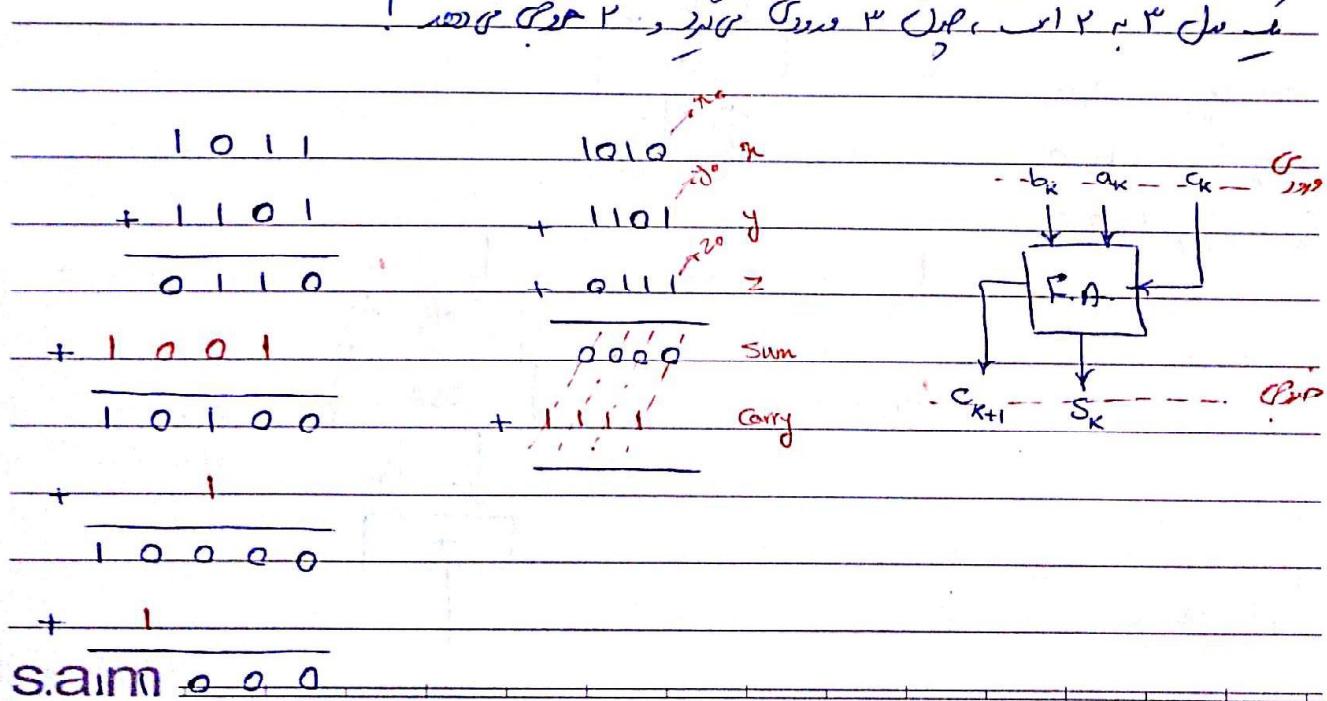
## Q. 14 Carry Select Adder

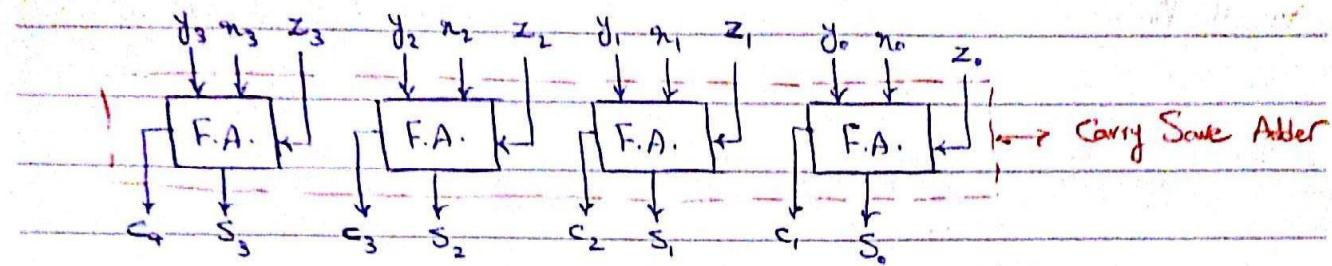
$$+ \begin{array}{|c|c|c|c|c|c|c|c|} \hline a_{15} & a_4 & \dots & a_{12} & a_1 & \dots & a_8 & a_7 & \dots & a_4 & a_3 & \dots & a_1 & a_0 \\ \hline b_{15} & b_4 & \dots & b_{12} & b_1 & \dots & b_8 & b_7 & \dots & b_4 & b_3 & \dots & b_1 & b_0 \\ \hline \end{array}$$



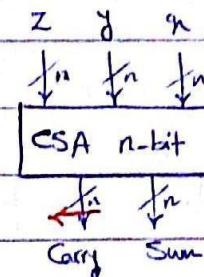
$$T_{RCA\ 4-bit} = 4 \times 2T_G, \quad T_{Carry\ Select} = T_{RCA\ 4-bit} + 3T_{MUX}$$

## ④ Carry Save Adder (CSA)



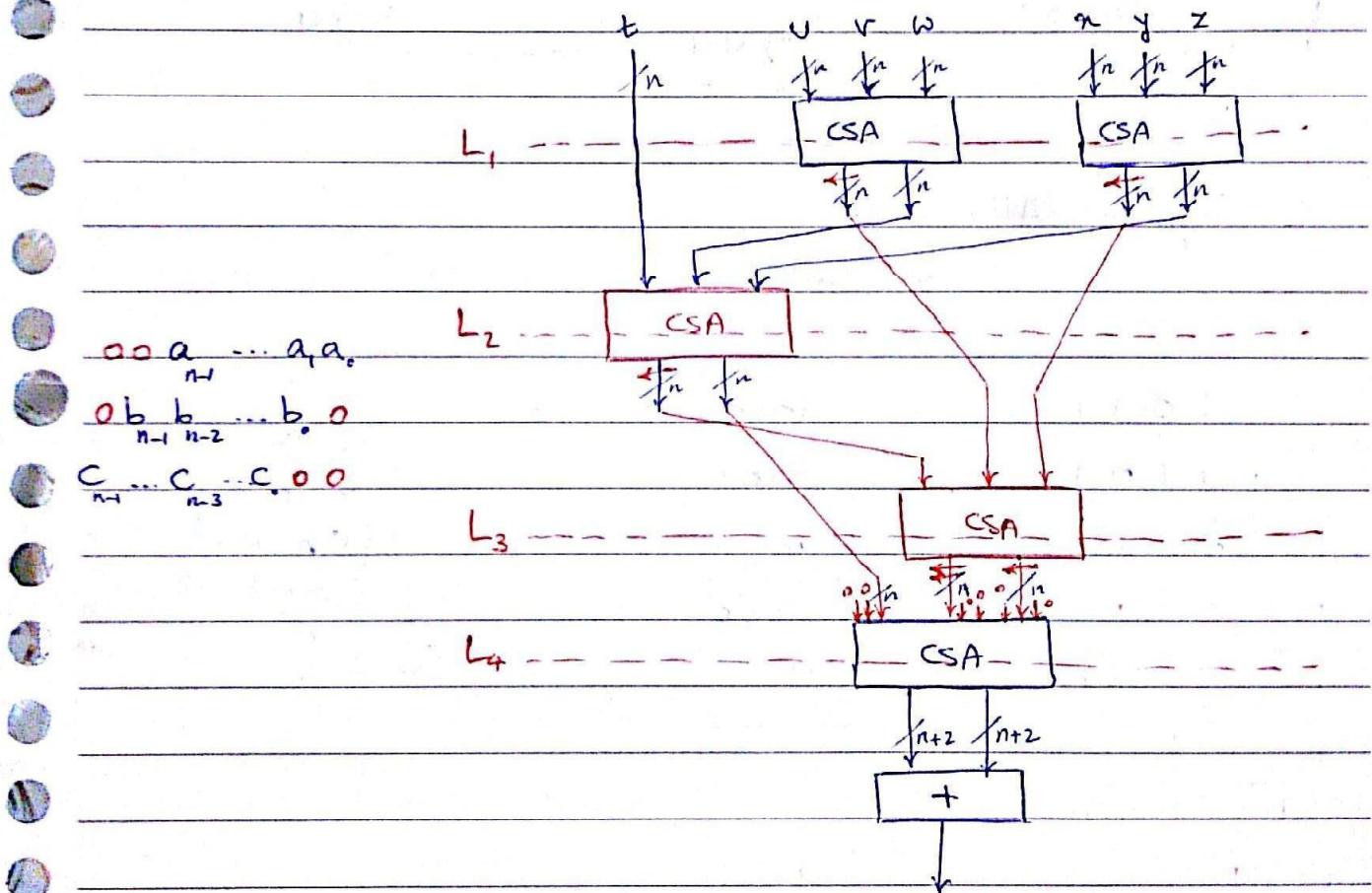


Carry Civil for chain addition ← Cok  
Sum Civil ↓

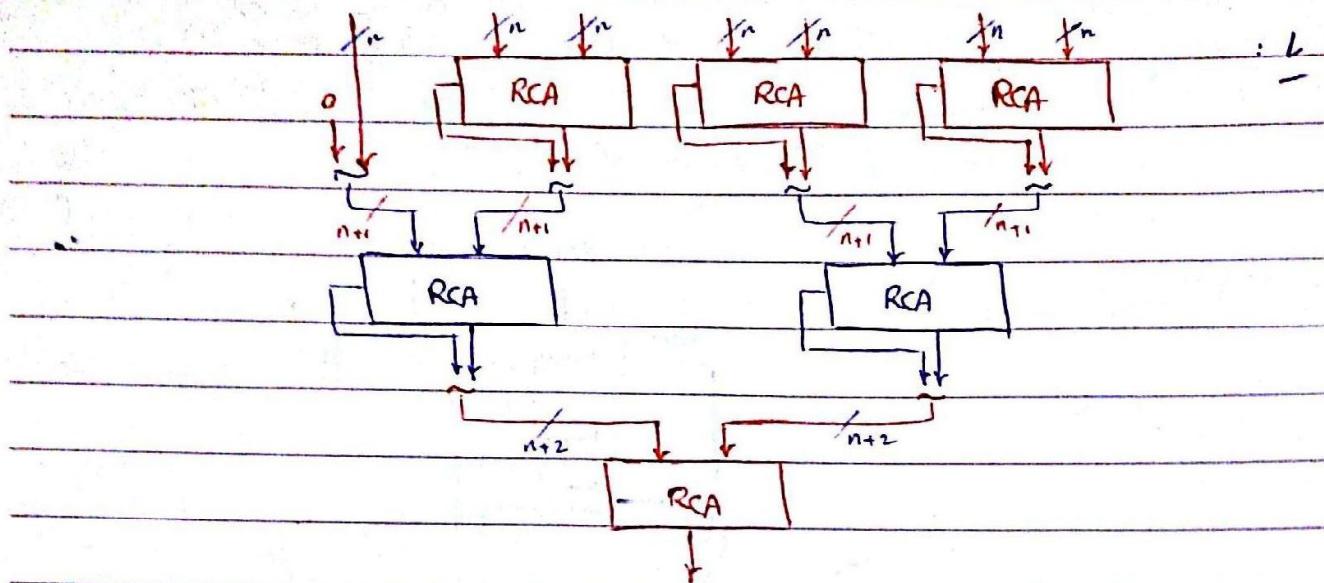


! CSA CSA : CSA in \*

t, u, v, w, n, y, z in n out v out \*



s.a.m



### ⑤ BCD Adder

3759  
 $(0011, 0111, 0101, 1001)_{BCD}$

$0011$

$+ 0101$

$\underline{1000} \rightarrow$  11 BCD

①  $\text{J}^{\text{w}}$

$0101$

$+ 1000$

$\underline{1101} \rightarrow$  11 BCD

②  $\text{J}^{\text{w}}$

$1000$

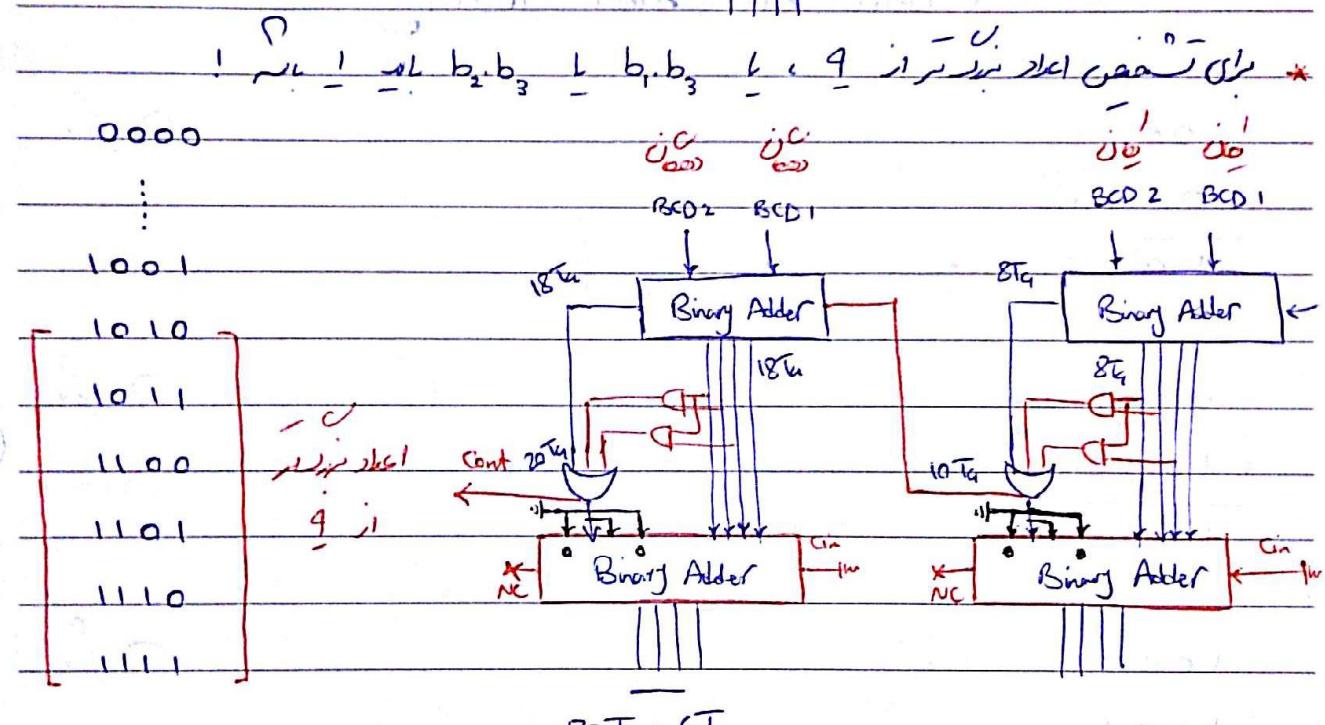
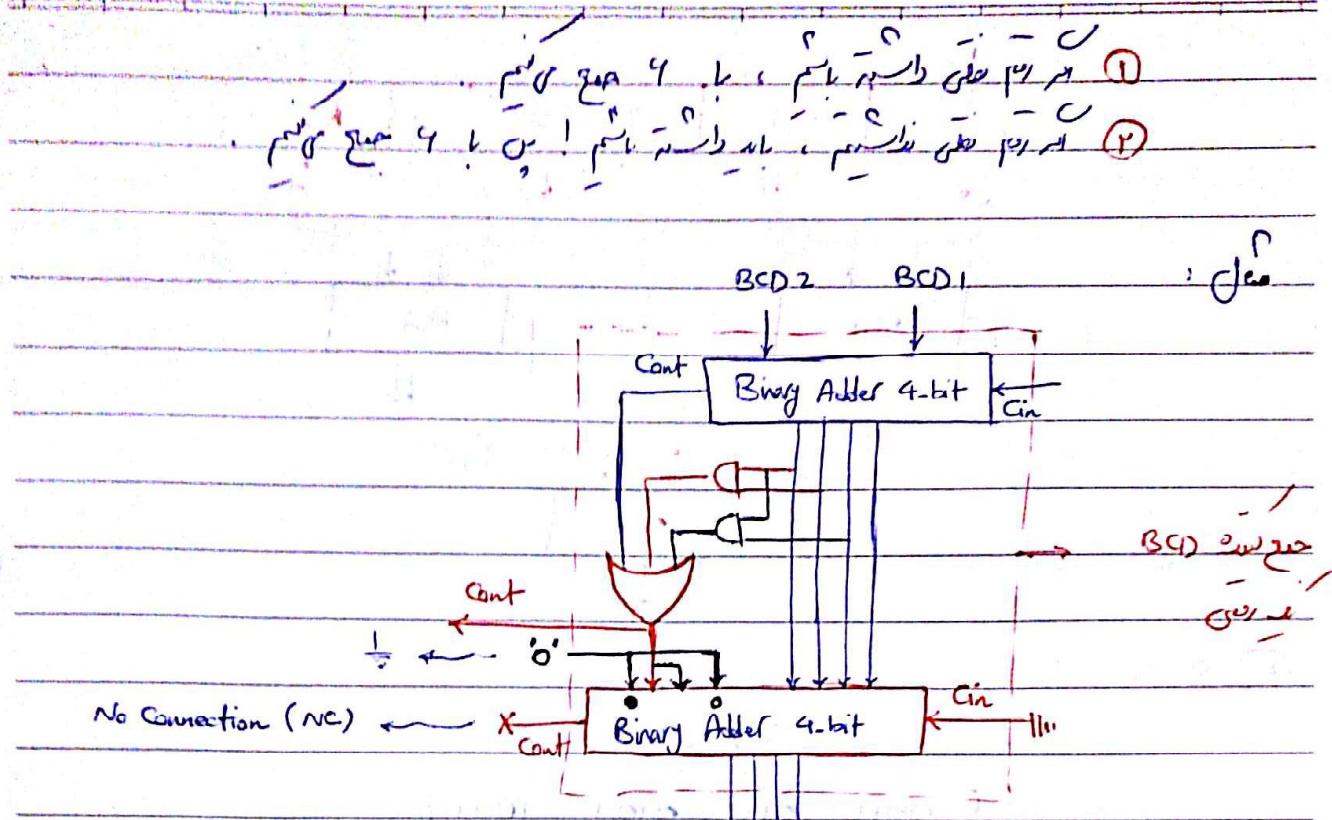
$+ 1001$

$\underline{0001} \rightarrow$  11 BCD

③  $\text{J}^{\text{w}}$

✓ 11 BCD

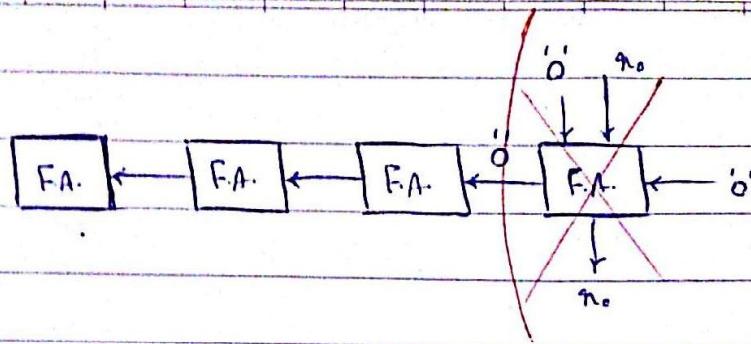
s.a.m



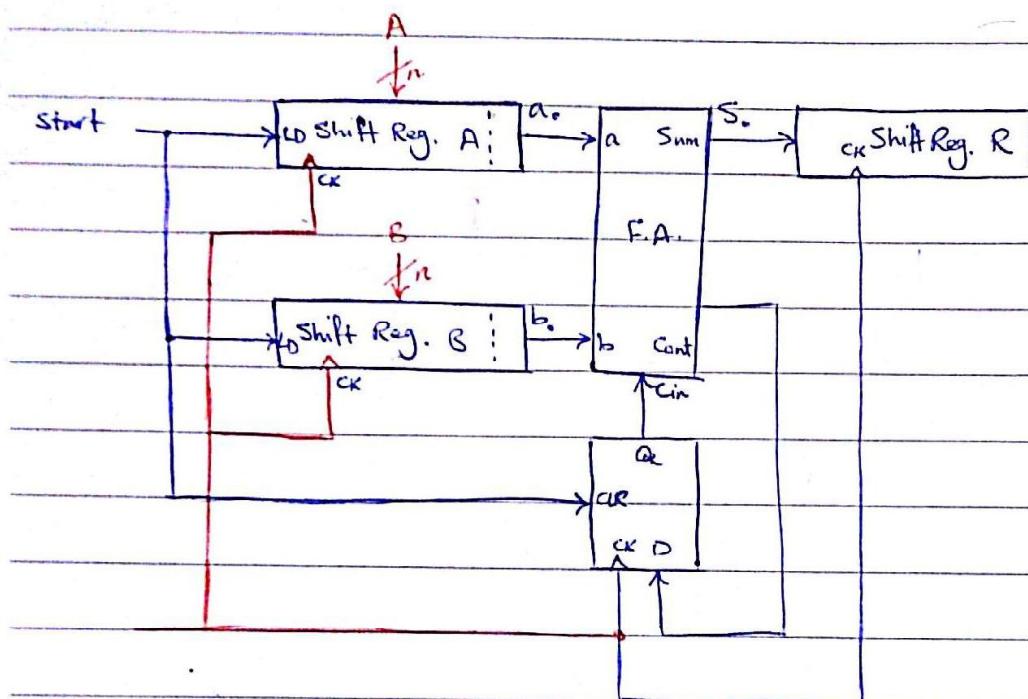
$$T_{\text{BCD Adder}} = m \times 10T_q + 6T_q$$

$m$

s.a.m



### ⑤ Serial Adder



$$[N]_2 = 2^n - (N)_2$$

Ans J.S.

$$(N)_2 + [N]_2 = 2^n = \underbrace{100\cdots 0}_{n\text{-bit}} \equiv 0$$

$$\begin{aligned} (N)_2 + [N]_2 &\equiv 0 \\ (N)_2 + (-N)_2 &= 0 \end{aligned} \quad \left\{ \Rightarrow (-N)_2 \equiv [N]_2 \right. \quad O_5$$

s.a.m

$$(A)_2 - (B)_2 = ?$$

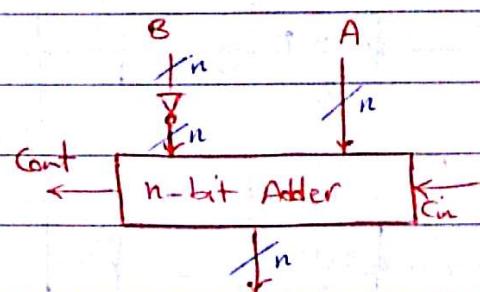
$[B]_2$  = 2's Complement of B

$$(A)_2 + (-B)_2 \equiv (A)_2 + [B]_2 = ?$$

$(\bar{B})_2$  = 1's Complement of B

$$[B]_2 = 2^n - (B)_2 = \underbrace{(111\dots1)}_{(\bar{B})_2} - (B)_2 + 1 = (\bar{B})_2 + 1$$

$$(A)_2 - (B)_2 \equiv (A)_2 + (\bar{B})_2 + 1$$

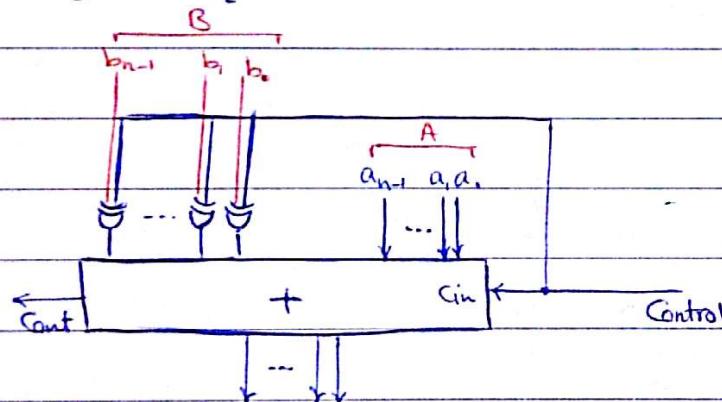


Q2, 18, V

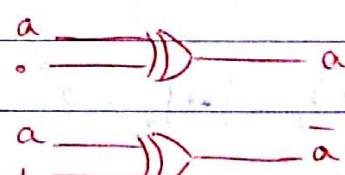
$$(A)_2 - (B)_2 \equiv (A)_2 + [B]_2 = (A)_2 + 2^n - (B)_2 = 2^n + ((A)_2 - (B)_2)$$

if  $(A)_2 \geq (B)_2$  : Cont = '1'

if  $(A)_2 < (B)_2$  : Cont = '0'



: out out / out out out



if Control = 0 : A + B

if Control = 1 : A + \bar{B} + 1 \equiv A - B

1010

SopJsc

$$\begin{array}{r} \times \\ 0110 \\ \hline \end{array}$$

$$A \times B = A \times (b_3^3 + b_2^2 + b_1^1 + b_0^0)$$

0000

$$+ \quad \begin{array}{r} 10100 \\ \hline \end{array}$$

$$= Ax^2 \times b_3^3 + Ax^2 \times b_2^2 + Ax^2 \times b_1^1 + Ax^2 \times b_0^0$$

$$+ \quad \begin{array}{r} 101000 \\ \hline \end{array}$$

$$+ \quad \begin{array}{r} 0000000 \\ \hline \end{array}$$

$$\hline \quad \begin{array}{r} 0111100 \\ \hline \end{array}$$

$$! z_A + - i^0 = - wP(jc)$$

$$2^K \times A = (K\text{-bit}) \text{ shift left}$$

$a_3 a_2 a_1 a_0$  : (Array Multiplier) SLR1 Row wise

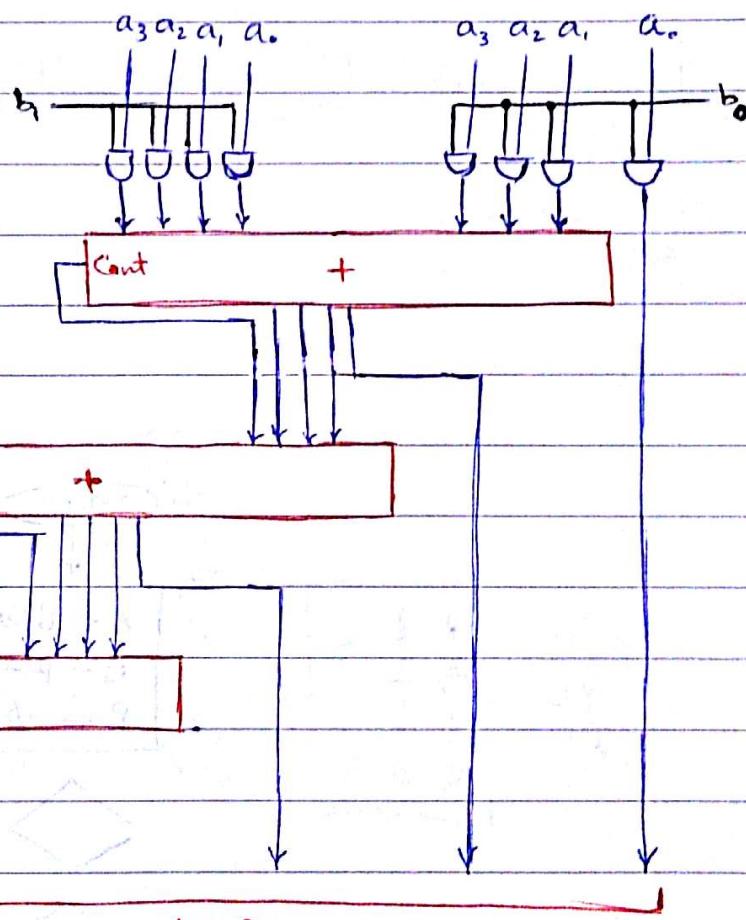
$$\times b_3 b_2 b_1 b_0$$

$$b_0 (a_3 a_2 a_1 a_0)$$

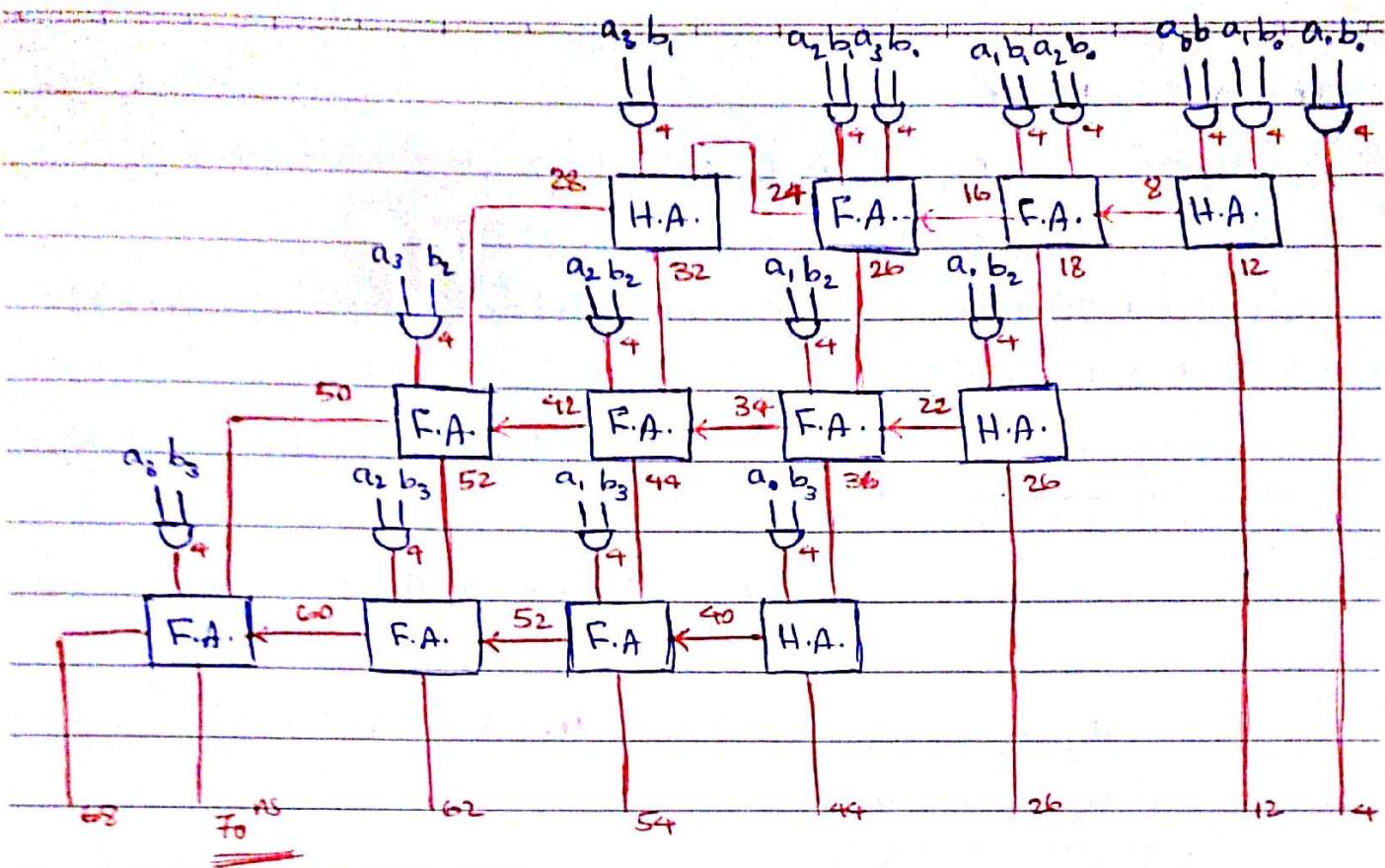
$$+ b_1 (a_3 a_2 a_1 a_0) 0$$

$$+ b_2 (a_3 a_2 a_1 a_0) 0 0$$

$$+ b_3 (a_3 a_2 a_1 a_0) 0 0 0$$

 $A \times B$ 

s.a.m

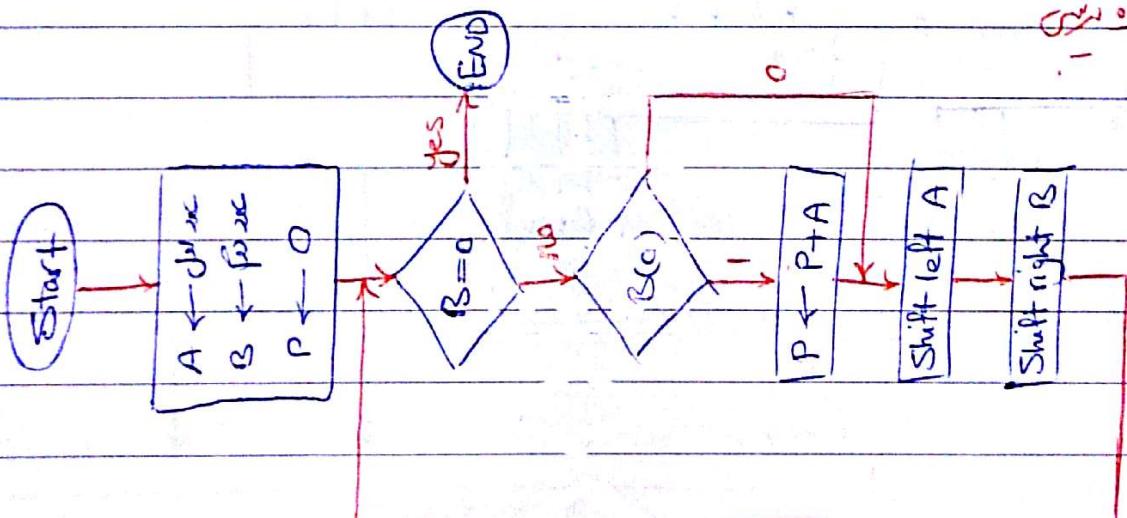


$c_0 = 1$ ,  $c_1 = 1$ ,  $c_2 = 1$

$$T_{\text{sum}_{FA}} = 10 \text{ ns}, T_{\text{cont}_{FA}} = 8 \text{ ns}, T_{\text{sum}_{HA}} = 8 \text{ ns}, T_{\text{cont}_{HA}} = 4 \text{ ns}$$

$$T_{AND} = 4 \text{ ns}$$

AND 2



s.a.m

Subject: A

Date:

$$A \leftarrow 1010$$

$$B \leftarrow 1011$$

$$P \leftarrow 0$$

1010 :  $\text{JLXc}$  :  $JW$

1011 :  $FJXc$

$$P \leftarrow 0 + 1010$$

$$P \leftarrow P+A$$

CK 1:  $A \leftarrow 10100$

shift left

$$B \leftarrow 101$$

shift right

$$P \leftarrow 1010 + 10100$$

$$P \leftarrow P+A$$

CK 2:  $A \leftarrow 101000$

shift left

$$B \leftarrow 10$$

shift right

$$P \leftarrow 1010 + 10100$$

$$P \leftarrow P+A$$

CK 3:  $A \leftarrow 1010000$

shift left

$$B \leftarrow 1$$

shift right

$$P \leftarrow 1010 + 10100 + 1010000$$

$$P \leftarrow P+A$$

CK 4:  $A \leftarrow 10100000$

shift left

$$B \leftarrow 0$$

shift right

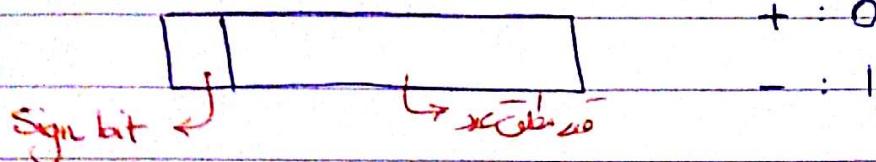
END

involves shifting bits:  $\text{JLXc}$

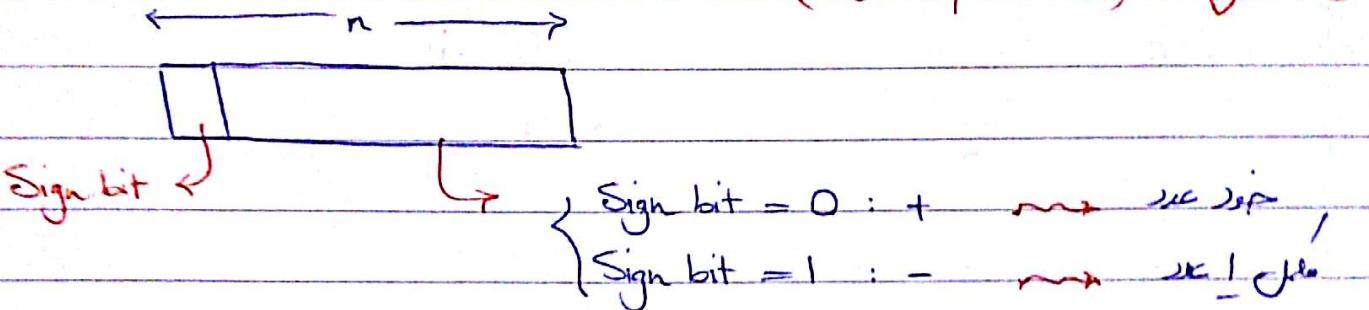
$\pm$  :  $b_{\text{out}}$

Bias (F)  $\rightarrow$   $b_{\text{out}}$  (R)  $\rightarrow$   $b_{\text{out}}$  (R)  $\rightarrow$   $b_{\text{out}}$  (I) :  $b_{\text{out}}$  shd be  $\times$   
s.a.m

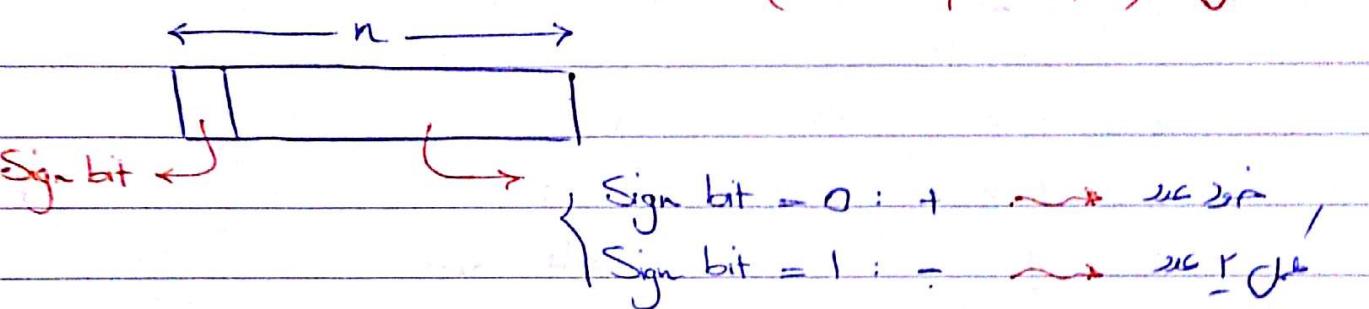
: (Sign - Magnitude)  $\bar{x}_0$   $\bar{z}_0$  ①



: (1's Complement) 1 job ④



: ( 2's Complement )  $\rightarrow$  ( 1's Complement )



$$+ 57 : \boxed{00111001} : \text{Jwb}$$

$$-87 : \quad +87 : 01010111 \xrightarrow[2's Comp.]{} \boxed{1}0101001$$

میں اکلہ سوتے دار، بس ارعن اور بس سوت جمع و تنویر عالی کا ۲ بنوں !

$$B = 8 + 4 + 1 \quad ( \quad B = 2B - B$$

$$13 = 16 - 2 - 1$$

$$-13 = 2^1 + 2^0 - 2^4$$

$$B = 2B - B$$

$$B = +5F : 2 \times B = 01110010$$

$$B = 00111001$$

$$0100 = 101 \dots$$

7<sup>b</sup> ✓

$$\downarrow \quad \curvearrowleft \quad 2^{\circ}$$

## Booth's Encoding

9a, 11, 12

(Booth's Algorithm) is considered as follows

$$A \times B = A \times (b_3 2^3 + b_2 2^2 + b_1 2^1 + b_0 2^0)$$

$$= (A \times 2^3)b_3 + (A \times 2^2)b_2 + (A \times 2^1)b_1 + (A \times 2^0)b_0$$

٣ - ٢ - ١ - ٠  
 بـ ٣ بـ ٢ بـ ١ بـ ٠  
 ٣ - ٢ - ١ - ٠

١) عدد اسلان به اهدت جمع و تصریح مطلق هم نیست

١٣) مَنْ هُوَ مَكِلُ حَمْعٍ، بَلَانْ حَمْعٌ يَسْتَهْلِكُونَ

$$\begin{array}{cccc}
 x_2^5 & x_2^3 & x_2^1 & \\
 \hline
 1-1 & 0-0 & 0-1 & 0-1 \\
 =0 & =0 & =-1 & =-1 \\
 \hline
 \end{array}
 \quad
 \begin{array}{c}
 \text{11001010} \\
 \hline
 0-1 \quad 1-0 \quad 1-0 \quad 0-0 \\
 \hline
 x_2^6 & x_2^4 & x_2^2 & x_2^0
 \end{array}
 \quad
 \begin{array}{l}
 -2^6 + 2^4 - 2^3 + 2^2 - 2^1 \\
 = -54
 \end{array}$$

$$Ax(-54) = Ax(-2^6 + 2^4 - 2^3 + 2^2 - 2^1)$$

$$= -2^6 \times A + 2^4 \times A - 2^3 \times A + 2^2 \times A - 2^1 \times A$$

! Job shift w, t &  $\frac{1}{\mu}$

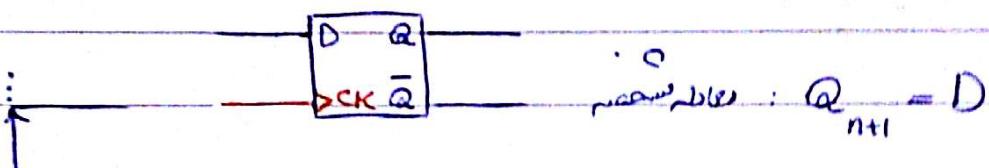
## ! Esb eset L r

۱۳- آنچه دارم!

## RTL (Register Transfer Level) Description / Design

- Registriertes Objekt: (Register) Objekt

or Register user



Cyclic Sym

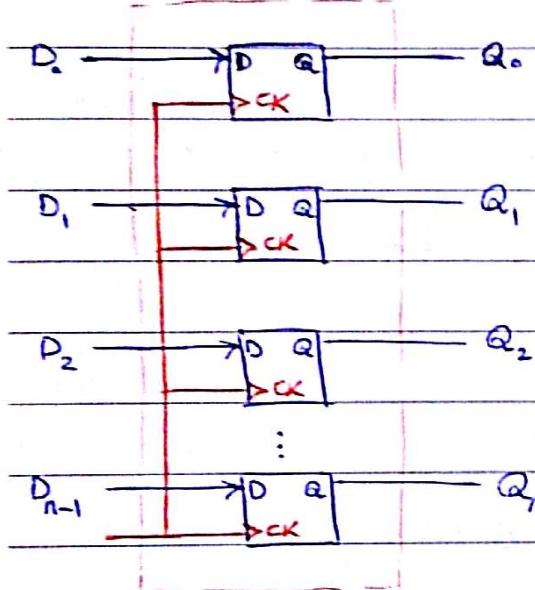
(Rising Edge)

## نے Register کیا ہے؟

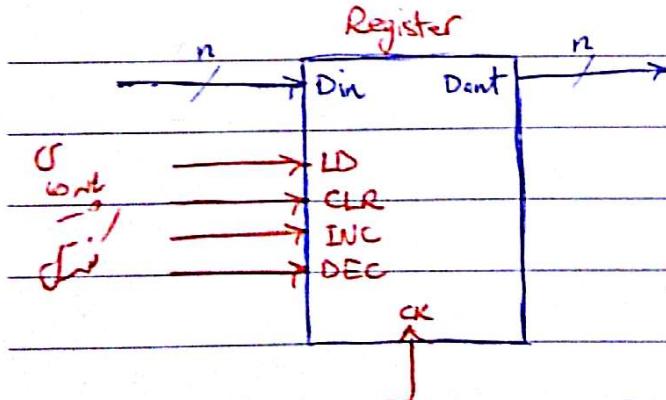
در دریا و در رودخانه، امکانات ذخیره می‌شوند.

درو مانی حفظکم اسخویه باش ! ما من حذام

در زبان های متفاوت اینها را نمیتوانند!



- 5  
Job Register



$\tilde{f} \circ g_k \in \mathcal{B}_\alpha$

دنه برقه می سوار

## LD (Load)

Register (رегистر) دوافعه : بارگذاری (Load) \*

نحوه عملیات دخواسته دادن به Register است !

Clock Pulse نیز که بسیار سریع باشد !

(! زمان CK (جی))

## CLR (Clear)

Register (رегистر) دوافعه : فعال نمودن (Clear) \*

## INC (Increment)

Register (رегистر) دوافعه : همچون مقدار اضافه کردن !

## DEC (Decrement)

Register (رегистر) دوافعه : همچون مقدار کم کردن !

اولین دوافعه دستورات (RTL) (LDR) است از اینها برای ذخیره مقدار (S) در Register (R) است.

برای ذخیره مقدار (S) در Register (R) دو دستور داریم :

1. LDR (Load Register) : ذخیره مقدار (S) در Register (R)

2. STR (Store Register) : ذخیره Register (R) را در آدرس (S) ذخیره کنید

## (RTL Statements) RTL جملات

أو دعائين شعبيين

- 1. جملة دعاء بمعنى ①

(MOP) Micro-Operation ②

new\_val  $\rightarrow$  Register  $\xleftarrow{\text{Clock Pulse}} \text{new_val}$ : MOP دعاء

Clock Pulse (Clock Pulse)

أمثلة: ③

load ①

store ②

copy ③

shift ④

Register  $\rightarrow$  Register  $\xleftarrow{\text{Clock Pulse}} \text{new_val}$ : ①(new\_val  $\rightarrow$  new\_val : Copy operation)Destination  $\leftarrow R_2 \leftarrow R_1 \rightarrow$  Source

\* دعاء مخصوص لبيان حذف المحتوى

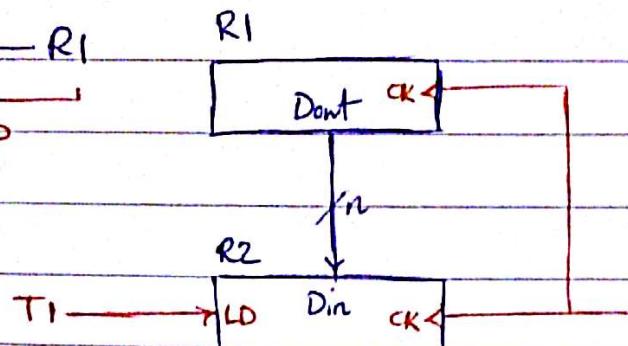
\* دعاء مخصوص لبيان نقل المحتوى

Clock pulse دعاء

Clock pulse دعاء

$T_1 : R_2 \leftarrow R_1$

map

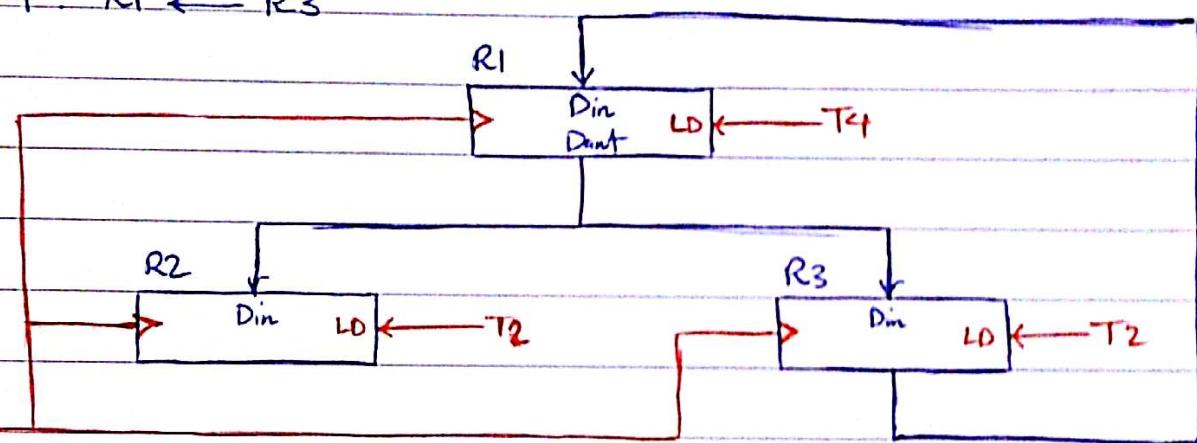


ulr  $T_1$  جزو  
 $T_2$  سوچه LD On

$T_2 : R_2 \leftarrow R_1, R_3 \leftarrow R_1$

$T_3 :$

$T_4 : R_1 \leftarrow R_3$



$CK$

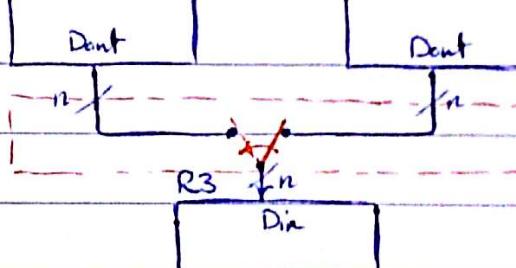
$T_0 : R_3 \leftarrow R_1$

$R_2$

$R_1$

JW

$T_5 : R_3 \leftarrow R_2$



(Internal BUS)

جزوی از مدار را باید با سیم کارهای دنده!

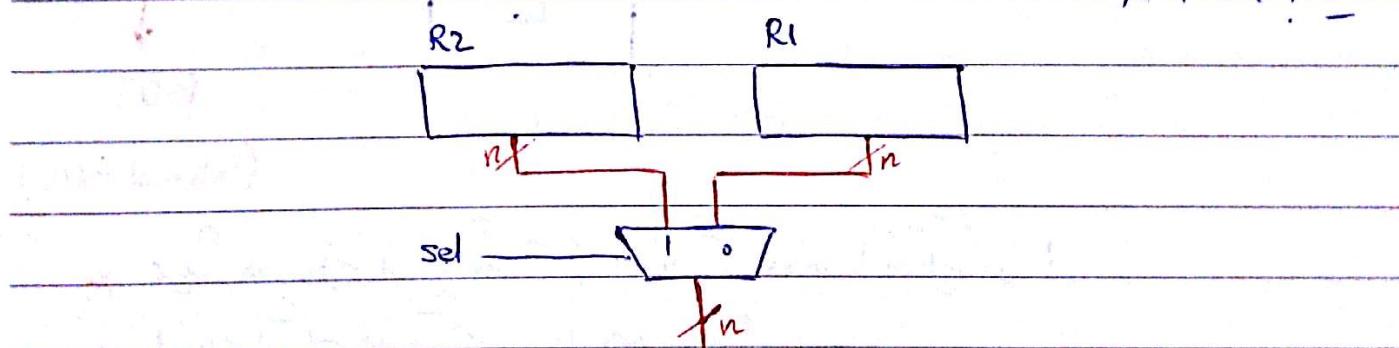
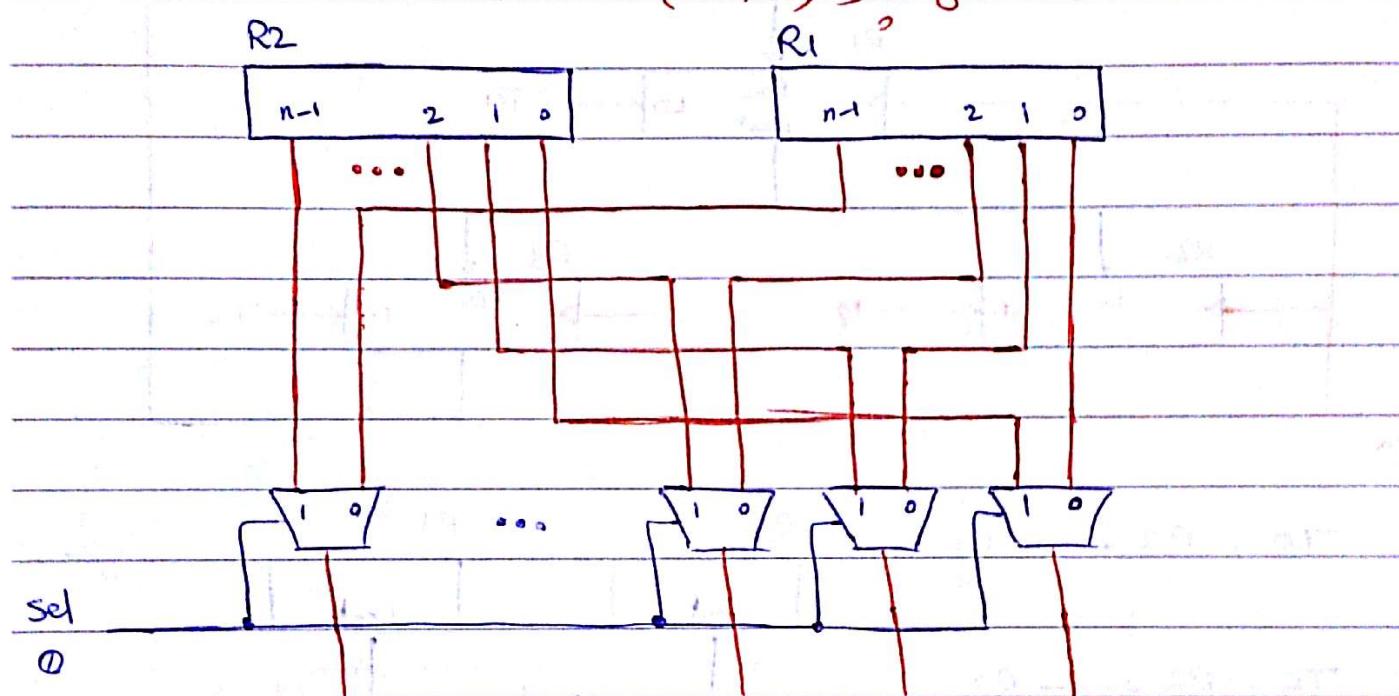
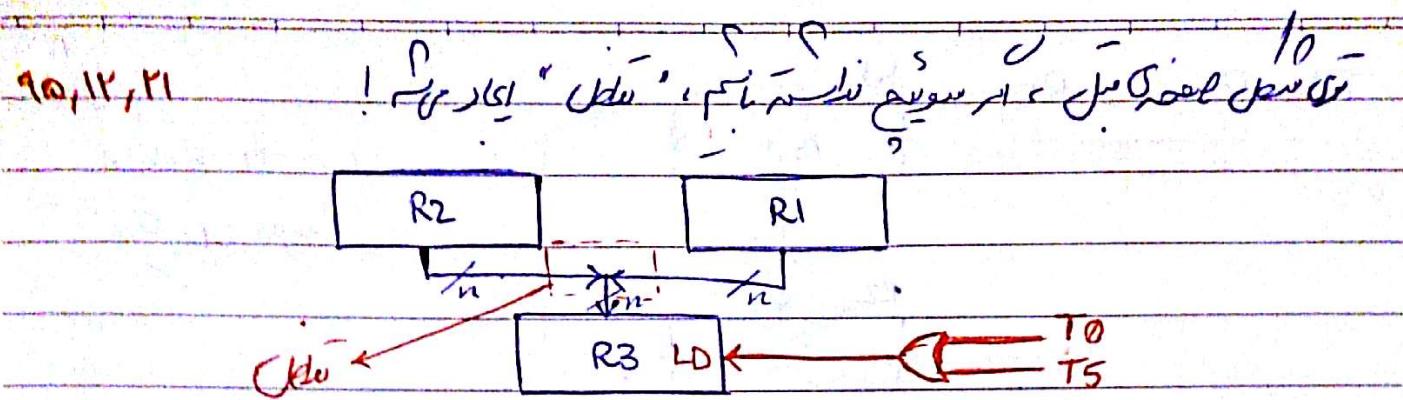
امان اعلی مسیر سیم کارهای دنده!

MUX یا چیزی! : این سیم کارهای دنده BUS اول!

s.a.m

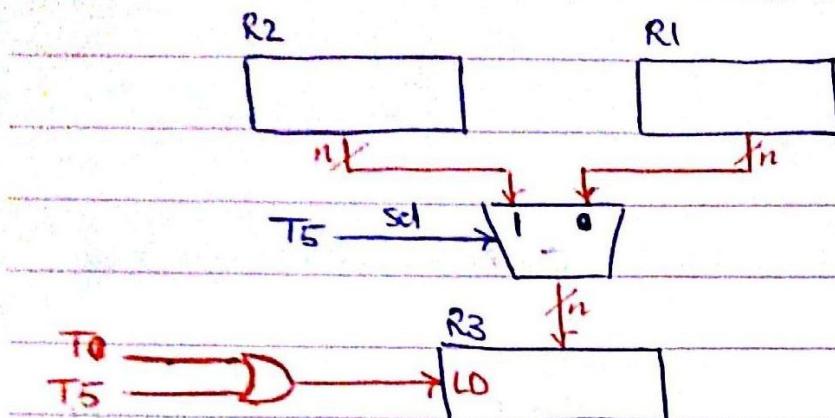
Subject: V4

Date:

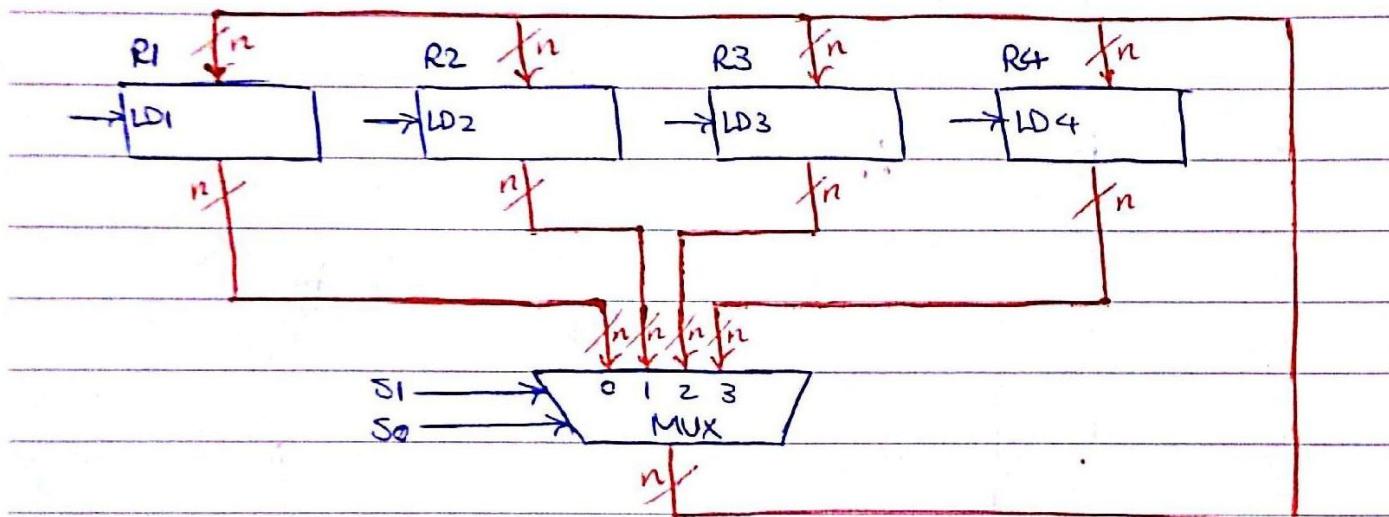


s.a.m

سالاری سهمیه ای و ملکی ۱۰٪



سل : طرق ارسال اطلاعات من مركب I2BUS ، من مركب ارسال از مرکب



$R3 \leftarrow R1$

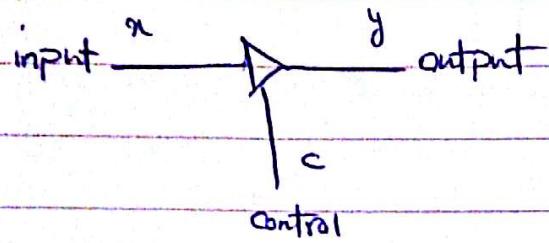
مکانیزم RI، MUX پرینتر (SISO: 00)

• سیگنال R3 را می‌باید در این مرحله MUX بپرس (LD3:1)

$$R2 \leftarrow R4$$

S.A.M نویسندگان R2 ، این جمله را در صورتی که MUX باید ( LD2 : 1 )

نحوه عمل جهاز.



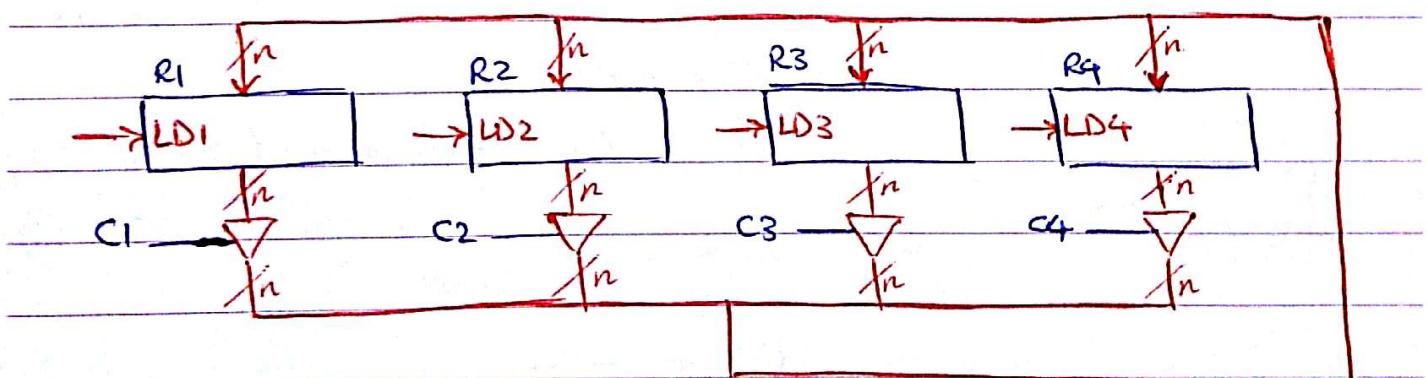
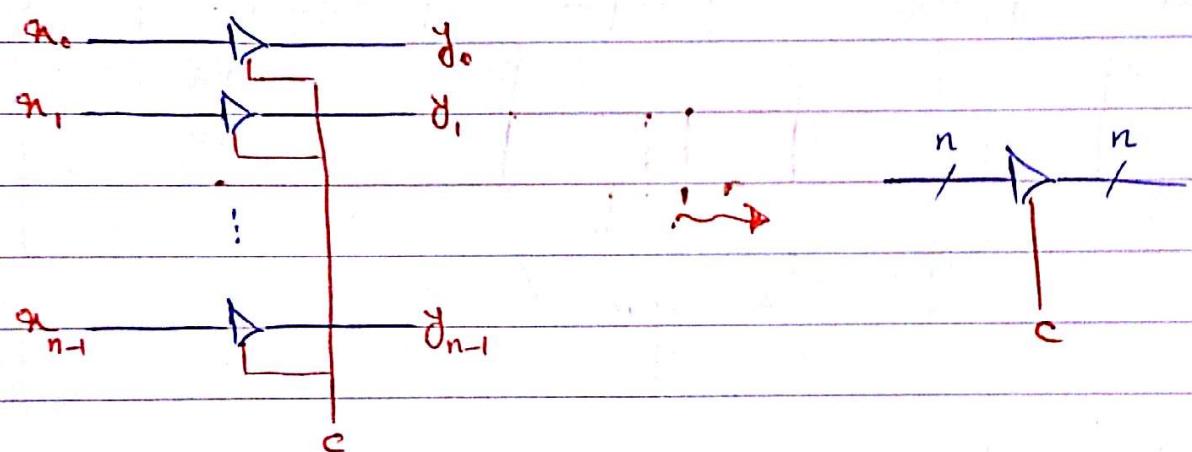
: بالشكل (س)

$c=0 \rightarrow 0 \rightarrow$  جهاز

$c \cdot x$	$y$
0 0	z
0 1	z
1 0	0
1 1	1

مخرج  
جهاز  
(جهاز).

$c=1 \rightarrow 0 \rightarrow$  جهاز



! ویر Data Path (Jewel) را پیش از \*

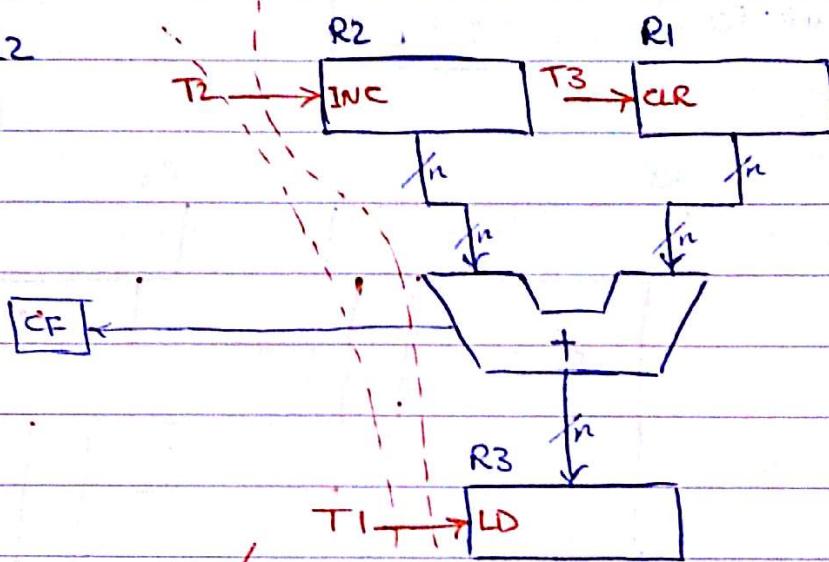
! ویر جعلی را به وکل و Register جه جعلی: Data Path \*

ویر جعلی را به وکل و Register جه جعلی: GPO, GMP (P)  
ویر جعلی را به وکل و Register جه جعلی: GPO, GMP (P)

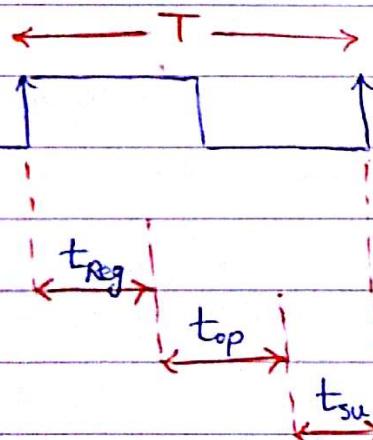
$$T_1: R_3 \leftarrow R_1 + R_2$$

$$T_2: R_2 \leftarrow R_2 + 1$$

$$T_3: R_1 \leftarrow 0$$



Control Unit / Control Path / جعلی ... Data Path



$t_{su}$ : setup time

$t_{reg}$ : ~~hold time~~

$t_{op}$ : ~~total time~~

$$f_{CK} = \frac{1}{T_{CK}} \quad (f_{CK}: CK \text{ (کیک))})$$

١٠٥ : مقارنة بين الـ XOR و الـ AND

الـ XOR يعطي ناتج عاً

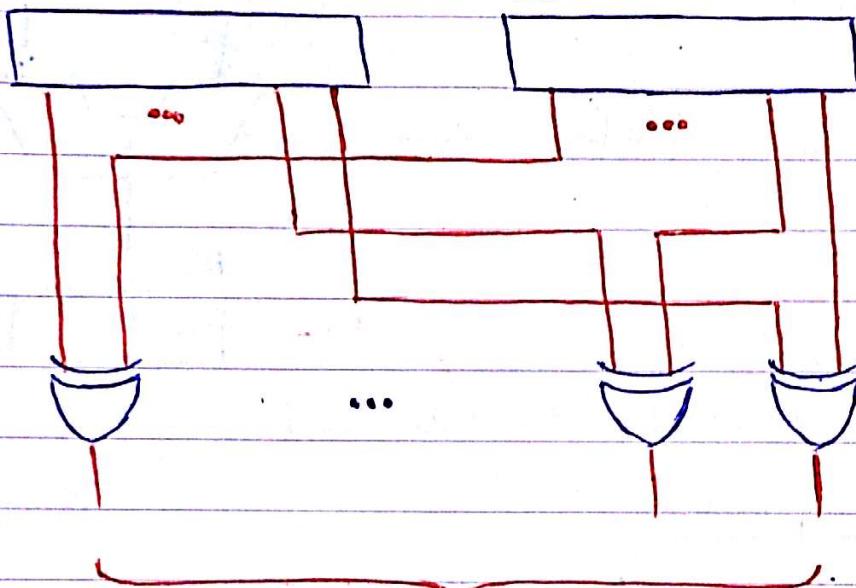
R1: 0 1 0 1 1 1 0 0

R2: 1 0 1 1 1 0 0 1

R1 XOR R2: 1 1 1 0 0 1 0 1

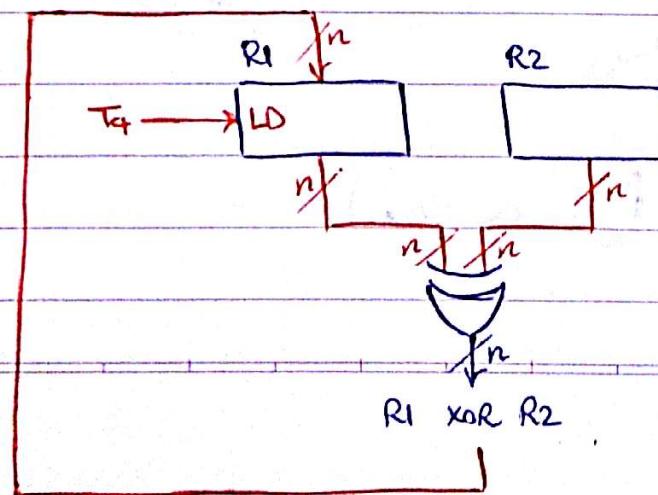
R1

R2



R1 XOR R2

T4: R1  $\leftarrow$  R1 XOR R2



s.a.m

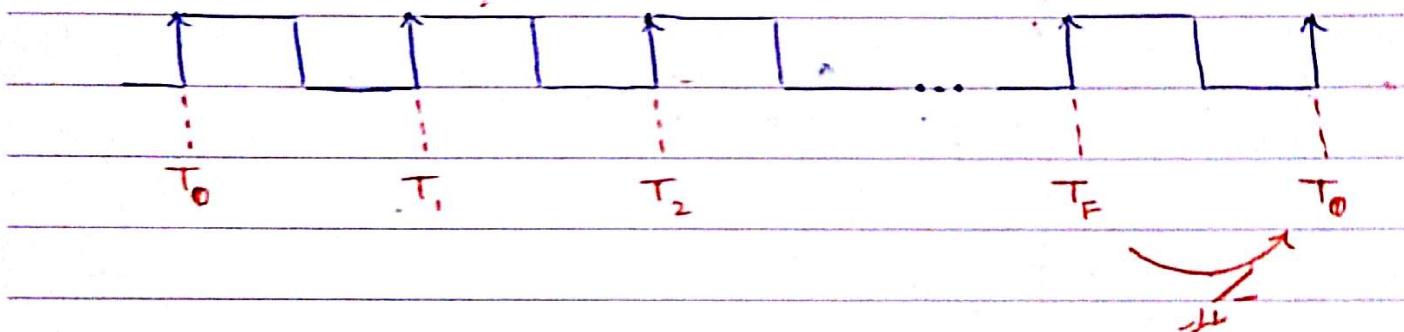
معلم سنبلي نهانی

\*  $T_1$ ,  $T_2$ , ... این ماده هایی دارند؟

لهم انت السلام السلام علیك السلام !

$$T_0 T_1 \cdots T_F \stackrel{?}{=} T_F T_0 \cdots T_1$$

$\uparrow$   
the  
 $\uparrow$   
it



\* ملک ایزدراز درجه  $T_0$  درجه سلسیوس در میان طیم. این مسازه

and  $F = \int d\omega \omega^2 |\psi(\omega)|^2$  is (counter)

!  $\sum \emptyset$  مدار

$$\emptyset \rightarrow 1 \rightarrow 2 \rightarrow \dots \rightarrow F$$

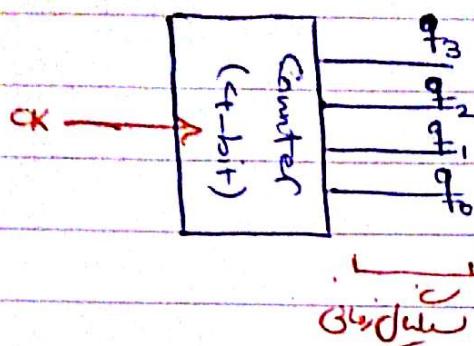
↑ \_\_\_\_\_ |

! fbiis gnt counter = !? nprf nsw 15 6 0 jklm h k w

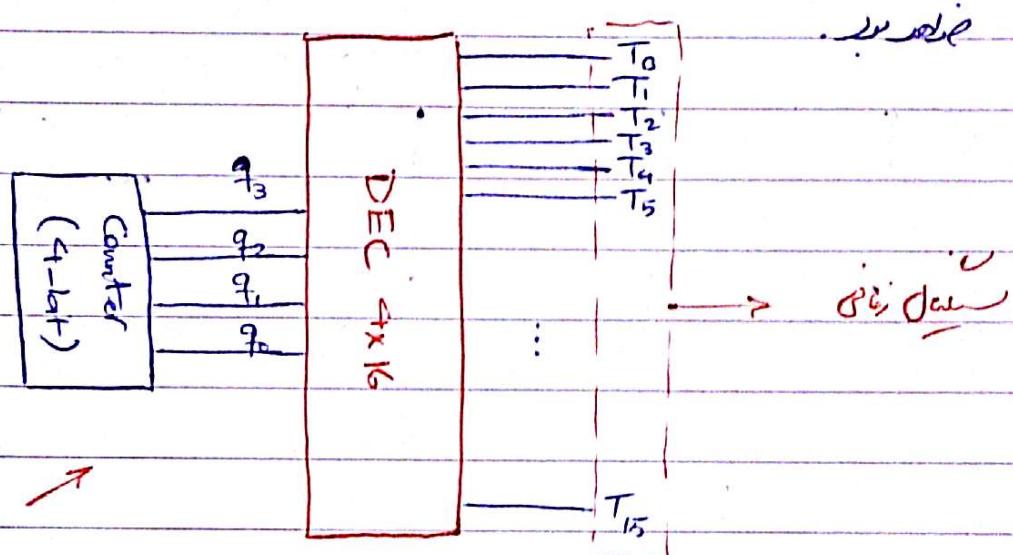
$0000 \rightarrow 0001 \rightarrow 0010 \rightarrow \dots \rightarrow 1111$

↑

## Binary Counter

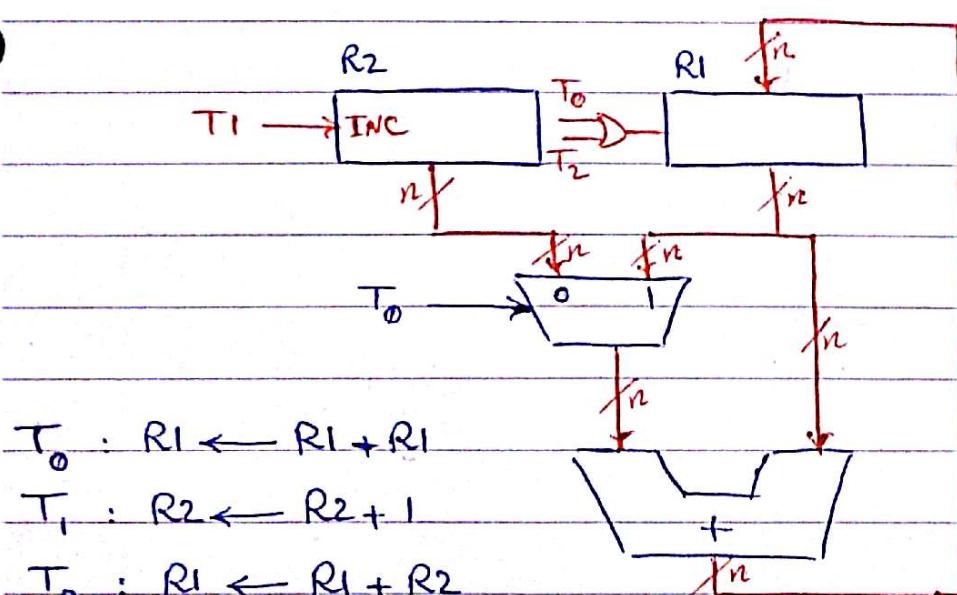


**سچنل:** سچنل نفع است نہ دنیاں میریم اے دنیاں ہے



مدار ترکیب سیگار نیزی

معلم : كمال مطر نور الدين العسلي و معلم RTL مصطفى عاصي



$$T_0 : R1 \leftarrow R1 + R1$$

$$T_1 : R_2 \leftarrow R_2 + 1$$

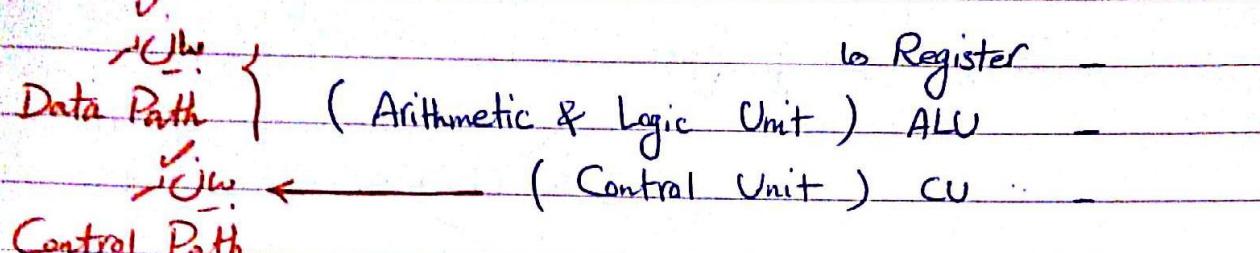
$$T_2 : R1 \leftarrow R1 + R2$$

RTL  $\rightarrow$  do it  $\underline{\text{first}}$   
(Analyze)  $\underline{\text{first}}$  !  
! message

99/1/19

CPU

Arithmetic &amp; Logic Unit



( General Purpose Register / GPR )

( Special Purpose Register / SPR )

: Internal Bus  $\rightarrow$  GPR \*

! تکیه می کنند ۱

! تکیه می کنند ۲

! SPR می باشد، این دو نوع را که هم در

دسترسی دارند، GPR و SPR \*

To SPR

( پردازنده ) Program Counter : PC ۱

( دستگاه ) Instruction Register : IR ۲

( داده ) Data Register : DR ۳

( آدرس ) Address Register : AR ۴

S.A.M.

- (نیز سرمهی ایجاد کننده) Stack Pointer : SP ⑥
- (نیز سرمهی ایجاد کننده) Program Status Word : PSW ⑦
- (نیز سرمهی ایجاد کننده) Temporary Register : TR ⑧

برای اینجا میتوانید متن مذکور را در اینجا ایجاد کنید.

### CPU کارکرد و دستورات

در این مرحله، سرمهی ایجاد کننده برای ایجاد سرمهی مسدود شده است. Fetch ①

در این مرحله، سرمهی ایجاد کننده برای ایجاد سرمهی مسدود شده است. Decode ②  
برای سرمهی ایجاد کننده برای ایجاد سرمهی مسدود شده است!

برای سرمهی ایجاد کننده برای ایجاد سرمهی مسدود شده است!

برای سرمهی ایجاد کننده برای ایجاد سرمهی مسدود شده است! Execute ③

: to SPR

برای ایجاد سرمهی ایجاد کننده برای ایجاد سرمهی مسدود شده است! PC  
برای ایجاد سرمهی ایجاد کننده برای ایجاد سرمهی مسدود شده است!

برای ایجاد سرمهی ایجاد کننده برای ایجاد سرمهی مسدود شده است! هر سرمهی ایجاد کننده برای ایجاد سرمهی مسدود شده است!

العنوان والبيانات من الذاكرة : IR

بيانات من الذاكرة : DR

بيانات من الذاكرة : AR

بيانات من الذاكرة : TR

بيانات من الذاكرة : DR و TR \*

بيانات من الذاكرة : SP

(Top of Stack / TOS)

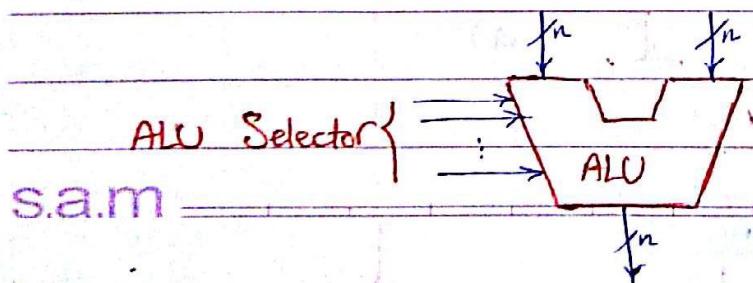
بيانات من الذاكرة : PSW

(Flag Register)

بيانات من الذاكرة : GPR

: ALU

Rotate , Shift , ALU



(o,w) من اجل مenge نوچه ایم Selector بولج +  
..... Shift

: ALU , to Register (جواہر)

: جو کسی نوچه پر گزینہ کرے

(o,w) Accumulator (جواہر جعل ①)

to Register (جواہر جعل ②)

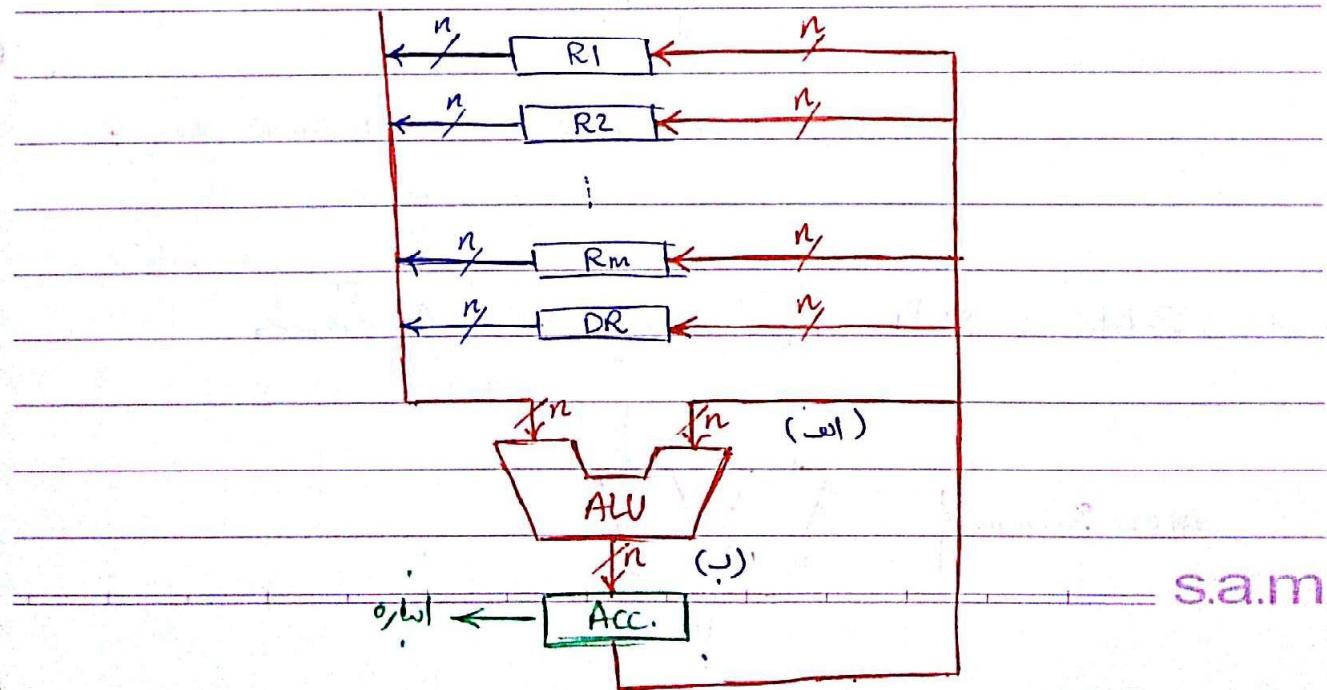
: Accumulator (جواہر جعل ①)

: جو کسی پر GPR کے Accumulator

..... سے جو ای Acc. کے نام دیا جائے (الل)

..... Register کے نام دیا جائے (الل)

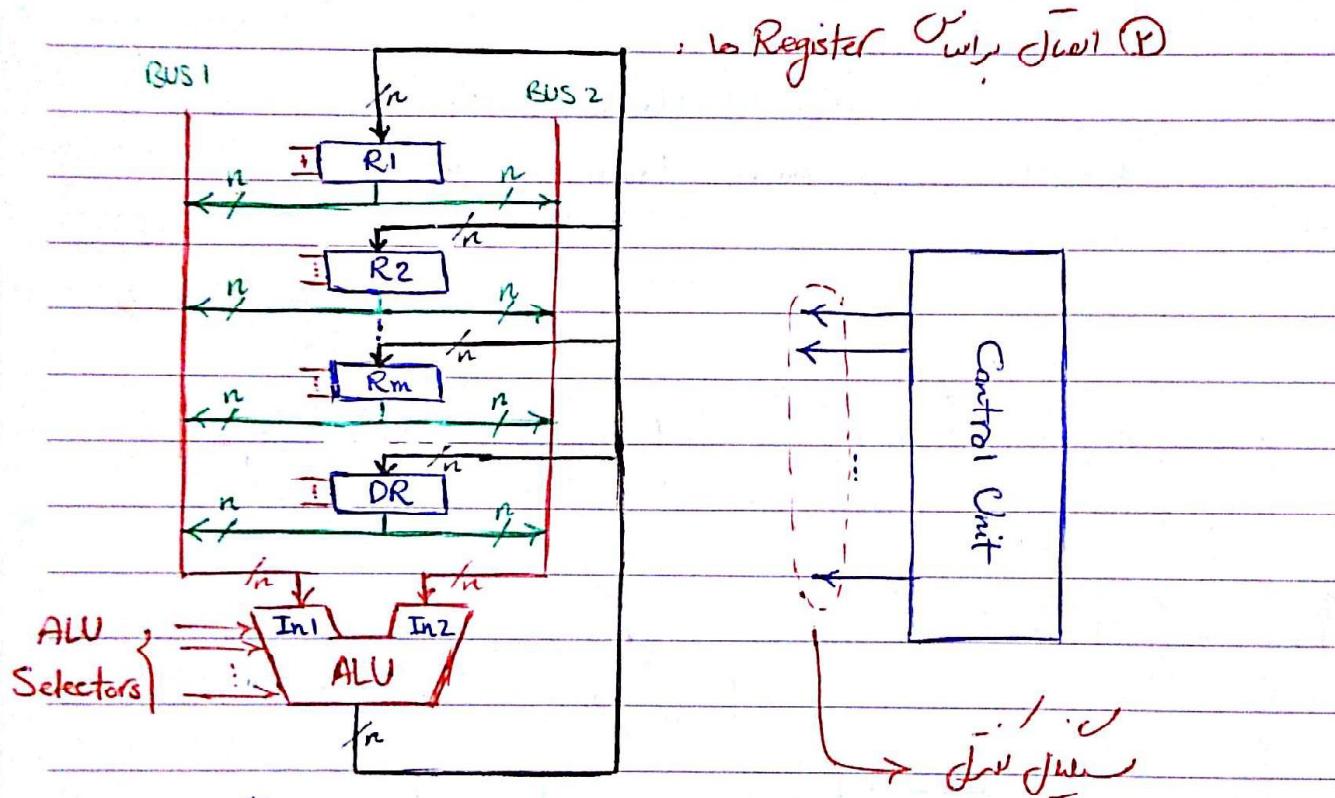
..... منظم



! ok (Slow in) slow :  $\rightarrow$   
! slow is slow. Will work in Acc. :  $\rightarrow$

$$R3 \leftarrow R1 + R2$$

- map 1:  $(Acc) \leftarrow R1$   $\rightarrow$   $VO$
- map 2:  $Acc \leftarrow R2 + Acc$   $\rightarrow$   $VOB$
- map 3:  $R3 \leftarrow Acc$



$$R3 \leftarrow R1 + R2$$

local w. BUS 2 goes to R2, BUS 1 goes to R1

! R3 job or now, ALU job is

s.a.m

\* این امر سبّت به از مدلی ، به کمّت اند سری نیاز نداشته (یعنی وتو !)

(CU) چه می خواهد که BUS 1 و BUS 2 مطالعه نماین ! نیز می خواهد که مطالعه نماین !

CU چه می خواهد که Execute ، Decode ، Fetch \*

کامپیوچر را ایجاد نماین !

: CU کامپیوچر را ایجاد نماین \*

(Hardwired) Hardwired ①

(Micro-Programmed) Micro - Programmed ②

: واحد کامپیوچر

- - - - -

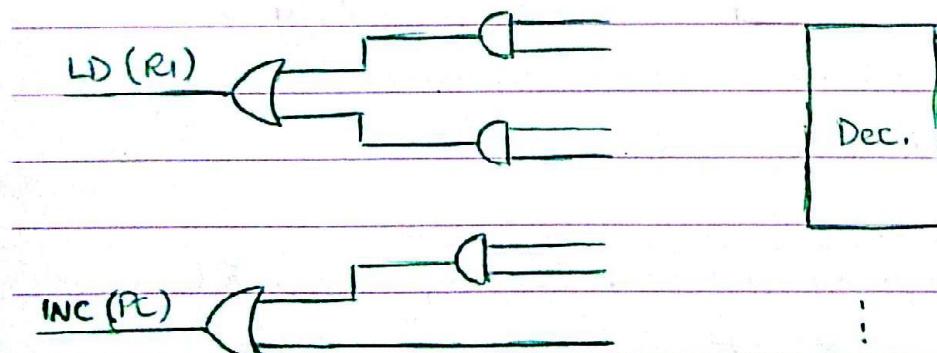
برای کامپیوچر های ساده ، معمولاً دستورات مجموعه ای دارند که معمولاً مجموعه ای دستورات هستند .

Decoder (دستورات را تجزیه کننده )

- - - - -

بلوک مرحله های دستورات . دستورات را در دستورات مجموعه ای دارند .

اداره اطلاعات . مطالعه نماین !



s.a.m

\* دلایل میتواند باشد، مثلاً محتوا نباشد.

: دلایل پردازش را در:

۱) اسایر از عملکرد برخطه لامپ است این سود، سعی در سود دادن  
نیز میتواند باشد، مثلاً نایاب باشد.

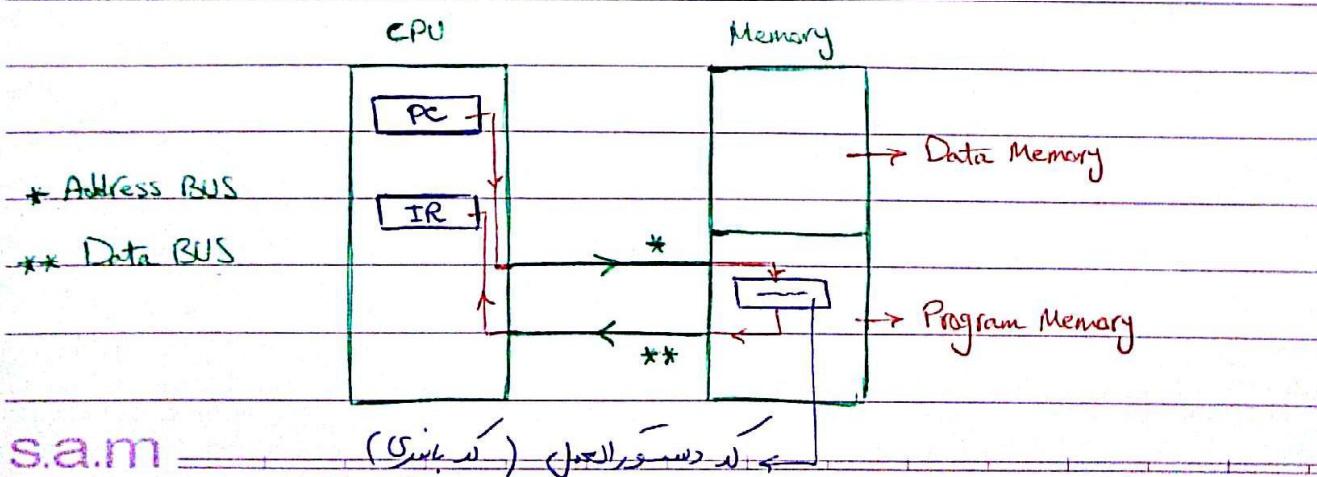
۲) این میتواند سعی در مطالعه ۱) را داشته باشد، این جایز نیست  
(حاقم بر زبانهای دیگر / زیرحاقم) نباید نیست.

۳) این میتواند سعی در دلایل میتواند باشد، مثلاً محتوا نباشد.  
نیز نباید نباشد و حاقم نیست، این برابر باشد و اطلاعات نیست نباید  
برخی بررسی.

• Hardwired و Micro - Programmed

• Hardwired و Micro - Programmed

• میتواند دلایل میتواند باشد



Fetch class Int

حاجة لـ PC لـ تطبيقات مثل Microsoft Word

دستور ایجاد و تغییر برای خلاصه در IR دستور ایجاد و تغییر برای خلاصه در IR

\* سعید ناطح طبل ، مل:

LD R2, R1

ADD R3, R2, RS

JMP 1000

انواع دستورات باینری به دو باندک بینک صفت (Binary Instruction CPU) می‌باشد:

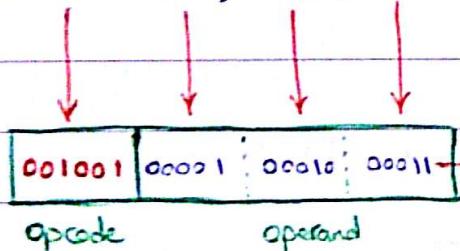
001010 00010 00001

16-bit

94, 1, 94

! Instruction Code / Job Codes

ADD r1, r2, r3 ( $r_1 \leftarrow r_2 + r_3$ )



Opsoed اپسید و اپسیڈنری اور اپسیند اور اپسیندری اور اپسیندنری

\* ملی ہر کمال - ملک اکمل خاں در تحریر نظریہ میسر ہے۔

Opcode-Field  $\geq \lceil \log_2 (\text{Number of Instructions}) \rceil$

--- shift، --- مجموع عددی، --- ساده طبقه بندی: --- ساده طبقه بندی \*

--- CPU  $\rightarrow$  Opcode ! --- CPU  $\rightarrow$  Opcode \*

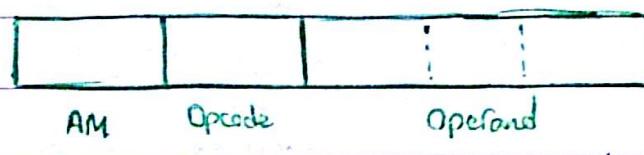
ADD R1, R2, 3 ( $R1 \leftarrow R2 + 3$ )



001001	000011	000100	000111
--------	--------	--------	--------

سؤال: --- CPU کس اور چھوٹے سے سمارہ 3 اور چھوٹے سے سمارہ 2 کی کیا کارکردگی ؟

پہلے ہی نہیں لگاتا کہ اس طرف کا ایسے مادہ میں کیا کیا کرنا ! اس کیسے Addressing (Mode) کا اشارہ میں اپنے کام کر دیتے وہ کیسے کہ اس کا اشارہ کر دیتے (Mode)



فہ ایسے سوال کیا کہ اس طرف کا ایسے مادہ میں کیا کیا کرنا !

: کام کرنے کے لئے اس طرف کا مادہ میں کیا کیا کرنا \*

I/O CS ③  
CPU CS ④

To Register CS ①  
by Hand ②

الواقع ببعض اجزاء مصر :

اداره اقتصادی : سمت پایین ۱

Cult. shock : Immediate / (متغير) مفاجئ (٢)

ریپ، I/O : Direct / ~~پرنس~~ ④

جائز : Indirect / غير مسمى ③

104, I/O : Reg. Indirect / ~~Supplementary~~ (Q)

: Auto-Increment or Auto-Decrement / جزء اضافي و مینیمی (4)

١٠٢ : Index / سچل

Tip : Relative / crie ①

جعی / Implied : ان طا حرا اس ! ۹

نحوی دسترسی را برای این هزار  $\text{ls}$  Operand :

If operand = 0

## - دسوار - صفر الرى

# operand = 1

## - دسخواں ملک اور میں -

# operand = 2

## دستورات (در اندیشی)

# operand = 3

## دستورات سعی اندیشی

ADD r1, r2, r3 ( $r1 \leftarrow r2 + r3$ )

سے ۱۰۵

ADD r1, r2 ( $r1 \leftarrow r1 + r2$ )

نحو احمد

ADD r1 , (AC  $\leftarrow$  AC + r1)

لـ (الضم)

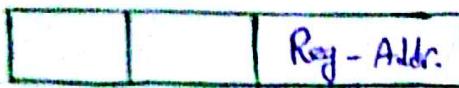
ADD سطراً تطبيقياً لـ ملخص مفهوم طرق (ست)

مختصر الدرس

:  $\lceil \log_2 N \rceil$  برای این سیستم باید  $N$  کویند

: Reg. AM /  $\lceil \log_2 N \rceil$  کویند ①

AM    opcode    operand



$$\lceil \log_2 N \rceil$$

Reg. File / Reg. Bank

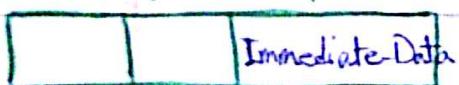
GPR 0	0
GPR 1	1
GPR 2	2
GPR N-1	N-1

Data

درین بخطیر استاره میدهند. بخطیر این در از حاصل داشتند،  
محضن: هنر اول در نهاد می خواهند اکام میانه و لب طبقه مودیلز نهاد تحقیق

: Immediate AM /  $\lceil \log_2 N \rceil$  کویند ②

AM    opcode    operand

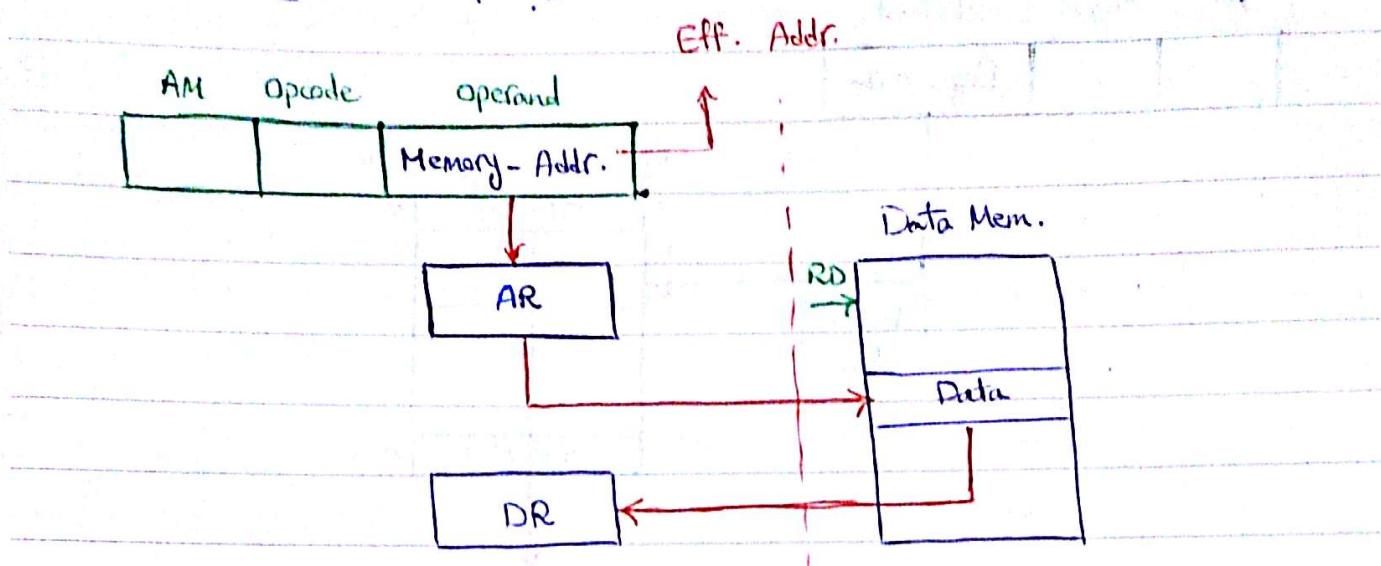


Operand می خواهند،  $\lceil \log_2 N \rceil$  کویند Data را می خواهند

\* CPU accesses to data immediately !

: Direct Am / paramagnetic  $\rightarrow$  ⑬

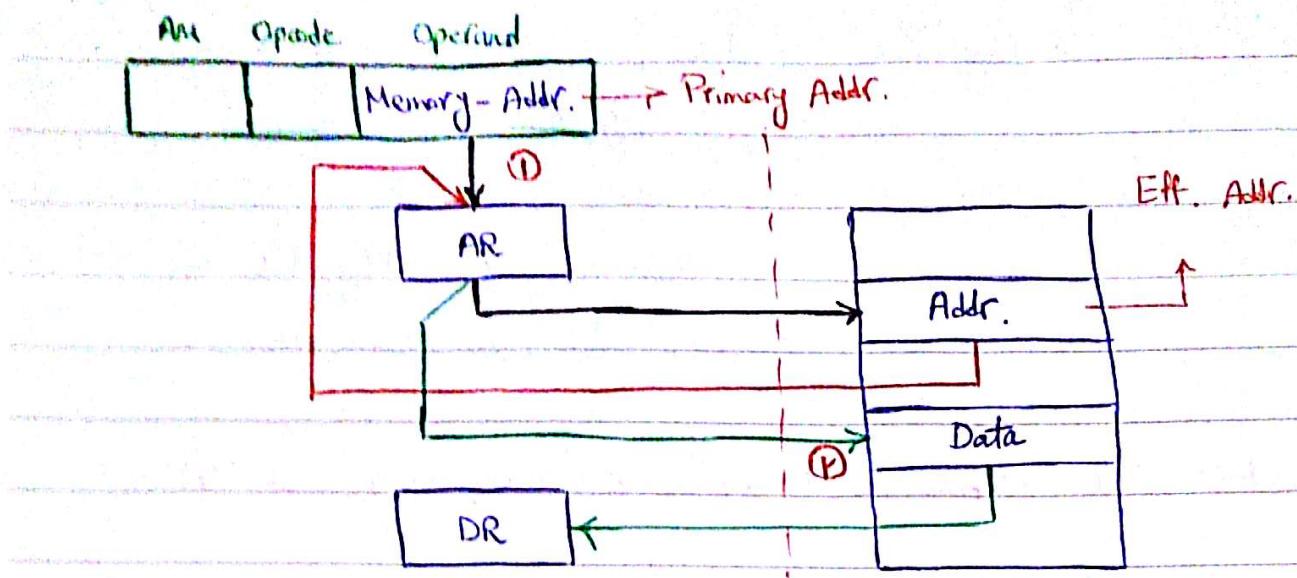
\* بلى حاتم ، I/10 استعاه منه ! (أيضاً بلى حاتم - تهنيئ داده !)



و محدود نموده . این Data حفظیت دارن DR نام دارند .  
و محدود نموده . این Data حفظیت دارن DR نام دارند .

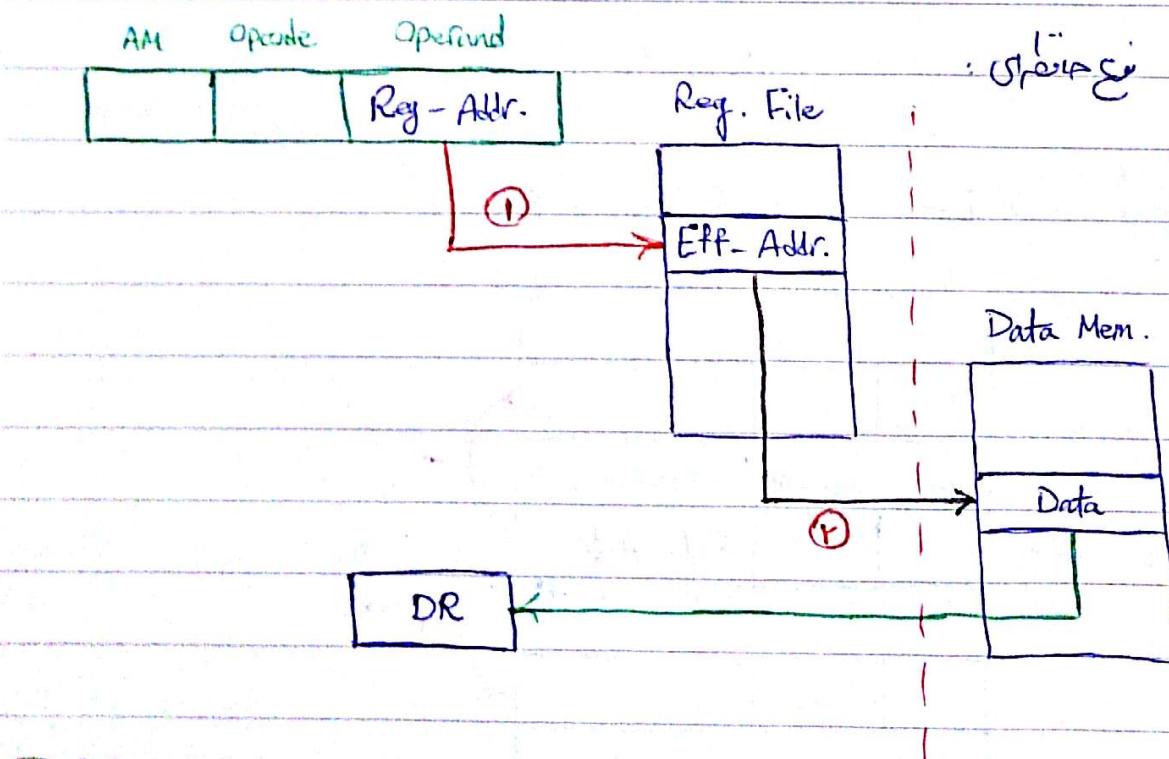
\* طبقاً لـ AR دون مemory-Addr. ملحوظة !

\* به اندیشی نه مسلطان  $\rightarrow$  اندیش معتبر نیاز است و همچنین  $\rightarrow$  اندیش معتبر (Effective)



Mem. Addr. چهارمین بخش از آدرس است که معمولاً در آدرس مبتدی (Data Segment) قرار دارد. Data Mem. را در آن می‌توان با آدرس AR پرداخت. Data همچنان که در آن قرار دارد، در DR ذخیره شود.

: Reg. Indirect AM / روش دسترسی از طریق Register @



اگر داده را Reg. File می‌خواهیم، این در Operand قرار نماییم  
اما ممکن است داده را Data Cst. نیز بنویسیم!

بروز روایتی از Reg. File \*

بروز روایتی از Data Mem. \*

### Auto-Increment or Auto-Decrement / اوت اینکرمیت یا اوت دیکرمیت ۴

Data Cst. نباید باشد! این در GPR Indirect شود  
GPR که Eff-Addr. باشد، در آن داده و نه این داده را بخواهد! همچنان که در آن داده را بخواهد!

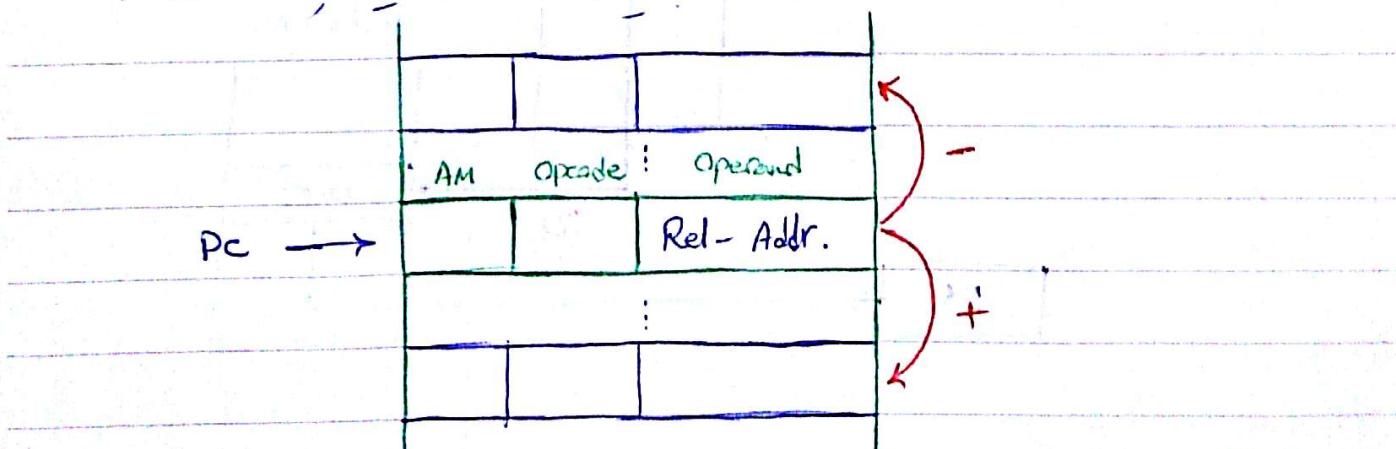
### Relative AM / امیج رلیتیو ۵

برای دسترسی از داده، آن را باز خواهیم داشت!

### Absolute / اolute

دسترسی از داده، Relative / امیج رلیتیو!

از آن که داده را باز نماییم!



? 2nd and 3rd row Eff - Addr. \*

$$\text{Eff-Addr.} = \text{PC} + \text{Relative Addr.}$$

\* میں کسی PC کو اپنے دستیکاری کے ترقیاتی سطح پر مل کر بخوبی کام کرنے کا طریقہ!

**مذمت** : موقتم للدبلوماسه الهرجانيه بکوام طبيعیتی !

: Index AM / ایندکس آم (۱)

17539	$\sin 0'$	17539 + 0
17540	$\sin 1'$	17539 + 1
17541	$\sin 2'$	17539 + 2
17542	$\sin 3'$	17539 + 3
:	:	
17629	$\sin 90'$	17539 + 90

Base      Index

Addr.

↑

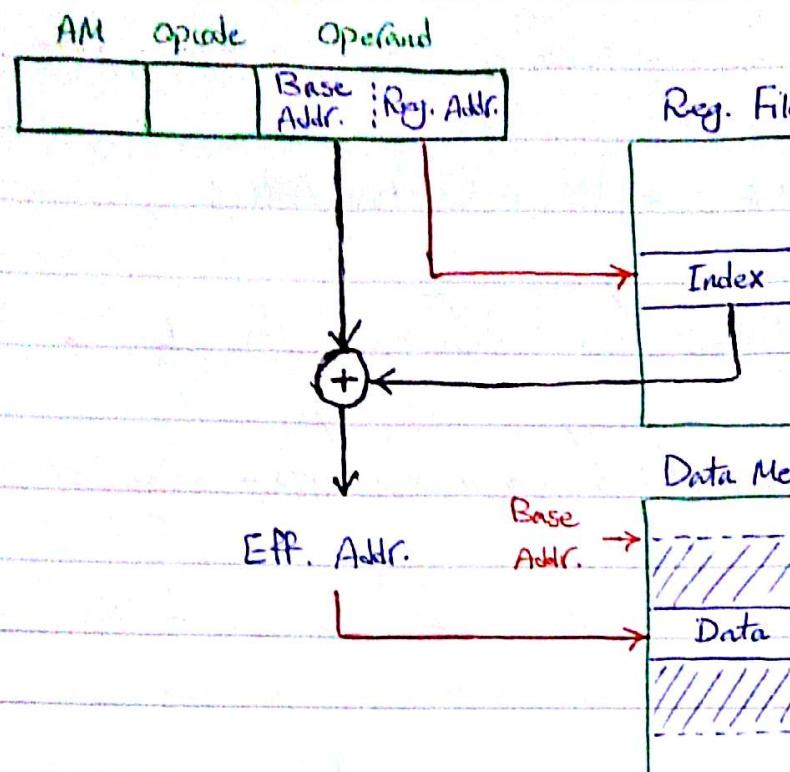
Sine

Table

↓

۱- میباشد تعریف داده و داده Data ها را بر اساس این دستگاه . این در اینجا ،  
۲- خود را در آگهی قرار داریم ، چون ساختن ما نیاز به است !

\* جعل المنهج كلام ادبي "Simile" او بحسب باليونیک "Base Addr. L" لـ "R" و "S" ، حفظ المنهج !



ادا داتا ميموري مدعنه ما دري و معمول است !

$\Rightarrow$  Direct Addressing, Index = 0 ✓ \*

$\Rightarrow$  Reg. Indirect Addressing, Base Addr. = 0 ✓ \*

: Implied AM /  $\text{Gelgen} \rightarrow$  ④

تمرين 8 (سرد و بطيء) : RET , CALL ، جملة امثل بجملة دوبرا

١) معاشر و معاشران

(Stack 8x32)  $\rightarrow$  Stack 1, 2, 3, 4, 5, 6, 7, 8 (P)

rib 8 :  $\begin{array}{c} \text{---} \\ | \\ \text{---} \end{array}$ ,  $\begin{array}{c} \text{---} \\ | \\ \text{---} \end{array}$ ,  $\begin{array}{c} \text{---} \\ | \\ \text{---} \end{array}$ ,  $\begin{array}{c} \text{---} \\ | \\ \text{---} \end{array}$ , Stack Pointer )  
(.  $\begin{array}{c} \text{---} \\ | \\ \text{---} \end{array}$   $\begin{array}{c} \text{---} \\ | \\ \text{---} \end{array}$   $\begin{array}{c} \text{---} \\ | \\ \text{---} \end{array}$   $\begin{array}{c} \text{---} \\ | \\ \text{---} \end{array}$  SP  $\begin{array}{c} \text{---} \\ | \\ \text{---} \end{array}$  .  $\begin{array}{c} \text{---} \\ | \\ \text{---} \end{array}$   $\begin{array}{c} \text{---} \\ | \\ \text{---} \end{array}$   $\begin{array}{c} \text{---} \\ | \\ \text{---} \end{array}$   $\begin{array}{c} \text{---} \\ | \\ \text{---} \end{array}$  )

94, 8, 14

(Pipeline) ~~reduces waste~~

ادیشن نہیں، کے مطابق (Performance) اس۔ ادنیں چھڑی بہرے عین ملکیت پر، معاشری بر عین احتجاج (اسمعان) ہے۔

بریت امر اور دنیا پر ہم تعداد میں بھی کم نہیں ہے۔ بڑا ہم زاری ہم بلیں ان مقام پر موجود ہو۔  
وہی اسم «بریت» بیان کیتے دخان «وسط ایسٹ» منسے! میں، زمانِ اجری دستی  
ور مطلع کرنے کے لئے۔

وَدَقَّتْ صَبَّارَةُ الْمَوْلَى (السَّوْلَى) فِي سَهْرِهِ، كَيْفَ تَلَاقَتْ حَدَّرَ (clock) هَذَا مَدْرَسَةُ حَرَقَّ سَعْدَ  
رَفِيعِ نَوْسَةِ

« طے احوالی جو درست اسی Clock ہے، میرے ساتھ اسی کا

( Clock Per Instruction / CPI )

برای این درسته همیشه clock نیاز است، مگر در طبع هر کجا برگزینش ها این خواسته است.

Jw

ADD R1, R2, R3

4 Clock

SUB R3, R2, R5

4 Clock

CALL SUB1

6 Clock

SUB1: AND R1, R1, 4

3 Clock

LD R5, 1000(R3)

8 Clock

RET

6 Clock

L inner clock

پس از این طبقه 31 بار زمان احتساب،  
 از مذکور شده 3 بار زمان احتساب داشته باشند!

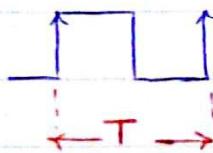
11. 200 میلیون هر دویا  $\times$  200 MHz, Clock Cycle 5 nsec  
 دارم.

sec

T

 $200 \times 10^6$ 

$$\Rightarrow T = \frac{1}{200 \times 10^6} = 5 \text{ nsec}$$



$$T = \frac{1}{f}$$

at all?

$$T_{Ex} = \text{inner clock} \times T_{clock}$$

L Execution Time

at Jw

$$T_{Ex} = 31 \times 5 \text{ nsec} = 155 \text{ nsec}$$

ذین اعلیٰ سنت و در علی مارسی است

• Clock دوچار کرنے والے Clock دی ①  
• Clock دوچار کرنے والے Clock دی ②

\* میزان این دستاں تغیر نموده، زمان اینکی تغیر نموده.

از این سه نتیجه باید مردمه کنیکل (Clock) محدوده مالی خود را تعیین کنند!

$$CPI_{av} = \frac{\text{Total execution time}}{\text{# Instructions}} = \frac{\text{# Clock Cycles}}{\text{# Instructions}}$$

$$T_{Ex} = CPI_{av} \times \#Ins \times T_{Clock}$$

~~Speed is not same~~ : Million Instruction Per Second / MIPS

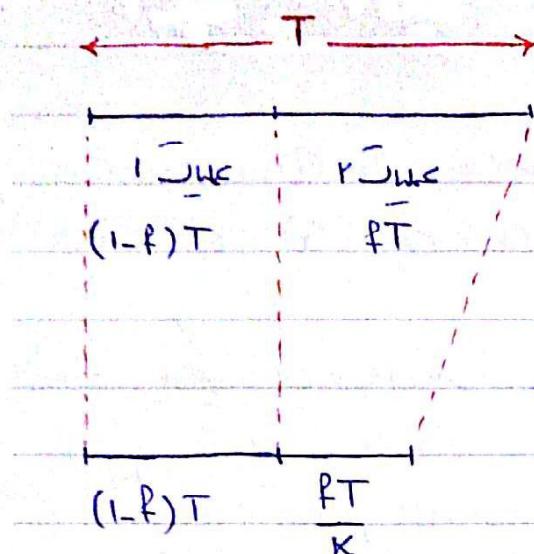
من ملک اجل سر؟ ما عالم سرک مالی اجل همینه!

$$\text{MIPS} = \frac{f_{\text{clock}}}{\text{CPI}_{\text{av}}}$$

MIPS \*  
میکرو پردازنده های MIPS معمولاً در سیستم های اندکاره هستند! مانند میکرو پردازنده های MIPS که با ۱۰۰ میلیون دستگاه اجرا نموده اند و میکرو پردازنده های MIPS که با ۲۰۰ میلیون دستگاه اجرا نموده اند!

## Amdahl's Law / Amdahl

فقط نسخه زبان اولیه برنامه T باشد و برای سه عبارت طبقه بانه ! طبقه می خواهد  
نمود . K بدلیل سرعت ترا اخراجی . هر جمله ممکن از این روش ساخت ( Speed up ) بر حساب نشود .



Speed up =

$$= \frac{T}{(1-f)T + \frac{fT}{K}} = \frac{1}{(1-f) + \frac{f}{K}}$$

—————  
مسوده بوده !!! \*

سال مانل Amdahl

براماري سابل ۲ کسند است: عبارت اعاده محاسب و عبارت حصر نامه. بحسب عبارت  
حصر نامه  $\frac{1}{K}$  دفعه زان اجرا نیافرود از خود اضافه می شود. تعدادی که عبارت نزدیک  
گشوده باشد باید سمع بر اجرائیم، اجل! برامار چون باید سمع بر خواهد بود؟

اعاده محاسب  
 $f = 0.7$        $1 - 0.7 = 0.3$

$K = 5$

$$\begin{aligned} \text{Speed up} &= \frac{1}{0.3 + \frac{0.7}{5}} = \frac{1}{0.3 + 0.14} = \frac{1}{0.44} \\ &= 2.27 \end{aligned}$$

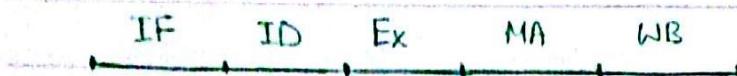
## : Pipeline (پلاین)

(۱) ساده

(۲)

(۳) پیچیده (مت‌آغاز بینی نیز شود)

## MIPS پردازشگر



## : Instruction Fetch (۱)

• PC را از میان مراحل (PC، ALU، ...) برگردان -

• PC را پاپ -

## : Instruction Decode (۲)

• نوع دستور را از میان دستورات مرسود (توصیه مادرنسل) -

• اسکارچ چشمها -

## : Execute (۳)

• از O&amp;I و ALU استفاده -

## : Memory Access (۴)

• داده را از حافظه اسکارچ -

## : Write Back (۵)

• از نتیجه ایجاد شده بازگشت -

\*: دستوراتی که در آن مولود نشوند (NP) هم دستوراتی هستند (NP-NP) \*

IF, ID, Ex, WB : دستورات

IF, ID, Ex, MA, WB : LD دستور

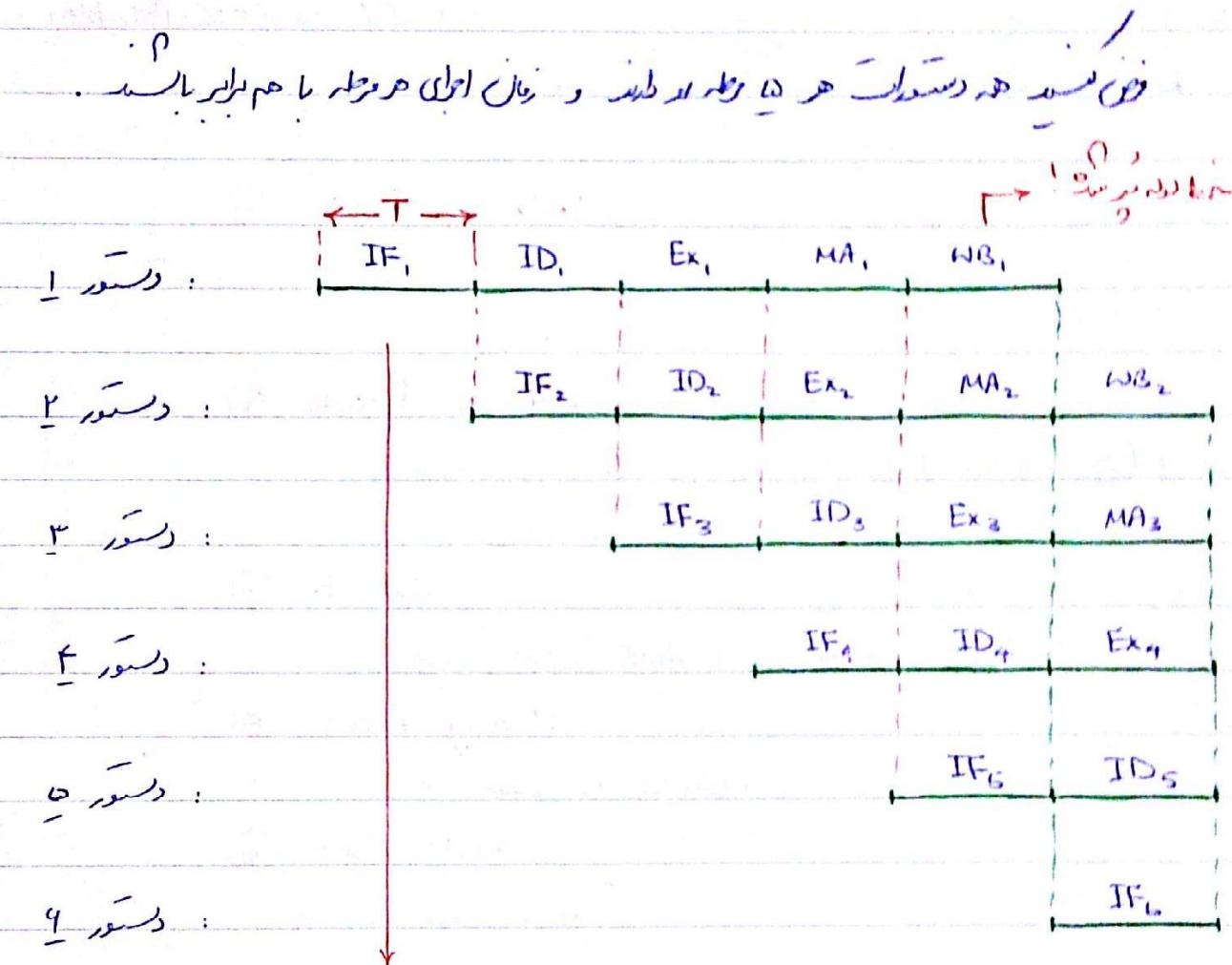
IF, ID, Ex, MA : ST دستور

IF, ID, Ex : bneq دستور

( IF - ID - Ex ) ! کام دستورها ، برای این طرف ! \*

1/ مرکز اعلیٰ دستورات زیر رسمی ! اس این نسبت مطابق سی Pipeline است . \*

Pipeline  $\equiv$  سی



Stage  
(n<sup>th</sup>)

WB<sub>2</sub> کام دستورها ! برای این طرف ، برای این طرف WB<sub>1</sub> ، برای این طرف \*

وَيُؤْمِنُ بِهِ الْمُرْسَلُونَ وَالْأَئِمَّةُ هُمُ الْمُرْسَلُونَ وَالْأَئِمَّةُ هُمُ الْمُرْسَلُونَ وَالْأَئِمَّةُ هُمُ الْمُرْسَلُونَ

$$\text{زمان اجرا} = \frac{\text{زمان اجرا}}{\text{دسته}} + \frac{\text{زمان اجرا}}{\text{دسته}} + \dots + \frac{\text{زمان اجرا}}{\text{دسته}} = 5T + (N-1)T$$

$$\text{number of nodes} = N \times ST$$

$\sigma = \sqrt{N} \text{ (جایی)}$

$$\text{Speed up} = \frac{\text{Actual time}}{\text{Ideal time}} = \frac{N \text{ (obs)} / \text{obs}}{N \text{ (obs)} / \text{obs}}$$

$$= \frac{n \times 5T}{(n+5-1) \times T} = \frac{5n}{(n+5-1)} \approx 5$$

$$\text{Speed up} = \frac{KN}{(N+K-1)} \quad (N \gg K)$$

$$\text{Speed up} = K$$

! ob Stage 11, Core IT über \*

! ↳ Stage  $\checkmark$  : K \*

نمره ۱: نعم نباید Pipeline مسادقه داد بلکه این به معنی از آن در پروژه‌نامه ایجاد نمایند / استفاده نیست باشد تغییرات متن ایجاد ننم! /

ستگر و پر طه انتها: و درستگر اول و دوستگر طه (Or Fetch) هم

دستور دم طرد مطابق فرستاده و دستور اول دستور فرستاده Fetch Unit، Decoding چنین دستوراتی که در حالت اولیه دستورات جایگزین شوند!

من این را در تقریب ساده نظری دارم که طبق این نظریه دستورات در مرحله Decoding به این شکاره از طریق آنها باید همان دفعه دوباره دستورات داشته باشند، ساده‌تر این دو دستورات کمتر و ساده‌تر دستورات دوباره باشند!

له می‌توان سهل این را دستورات واحد سینه دوباره جا نهاده که نهادن نموده! بلکه این متوجه خسارتی نموده تقریباً می‌تواند کمیستن Pipeline Reg. می‌شود.

و خسارتی خواهد داشت که نهادن این دستورات زمان زیادی از مطریک می‌برد و مطریک!

\* قیمت زمان  $T$  بوده اما اولین زمان نهادن  $T_{ov}$  است! (سریار نهادن!)

$$\text{Speed up} = \frac{N \times K \times T}{(N+K-1)(T+T_{ov})}$$

↑ overhead  
↓ کمترین زمان خطا دار

(1) ما فرض کردیم زمان خروجی کامپیوتر بالاتر است، اما طبق این نظریه از حسابات استفاده می‌کنیم که زمان خروجی کمتر است؛ به این رسمیت این نظریه می‌تواند کمتر است!

آن باید نیز زمان خروجی کمتر باشد! صنعت کامپیوتر طرد مطابقاً به طبقه نهادن این دستورات! پس بازهم زمان کمتر نخواهد بود!

94, X, 1

پایه ای از پایه

Stack

Flags

Stack

LIFO (Last In, First Out) Stack \*

Stack ← (Last In, First Out) LIFO - } پایه ای از پایه \*

Queue ← (First In, First Out) FIFO - }

Data چیزی است که از آن استفاده می‌کنیم؟ مثلاً این دستورات را می‌توان اجرا کرد:

- (Sub-Routine) کارهایی که انجام داده و خروجی داشته باشند.

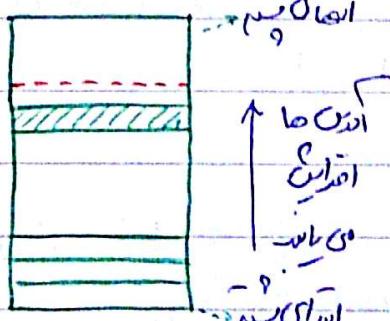
پایه ای از پایه دستوراتی که انجام داشته باشند \*

(دستورات خارجی) External / برون { ①

(CPU دستورات) Internal / درون { ②

برای این دستورات که درون کارهایی دارند، این دستورات را دستورات کارهایی می‌نامند.

برای این دستورات که خارجی دارند، این دستورات را دستورات خارجی می‌نامند.



SP : Stack Pointer

Stack  
(اشاره کننده)

سینه دهارت طبقه باشی: SP

- ۱) اخیر مطلق درسته استاده نه!
- ۲) اول مطلق خالی لایک استاده نه!

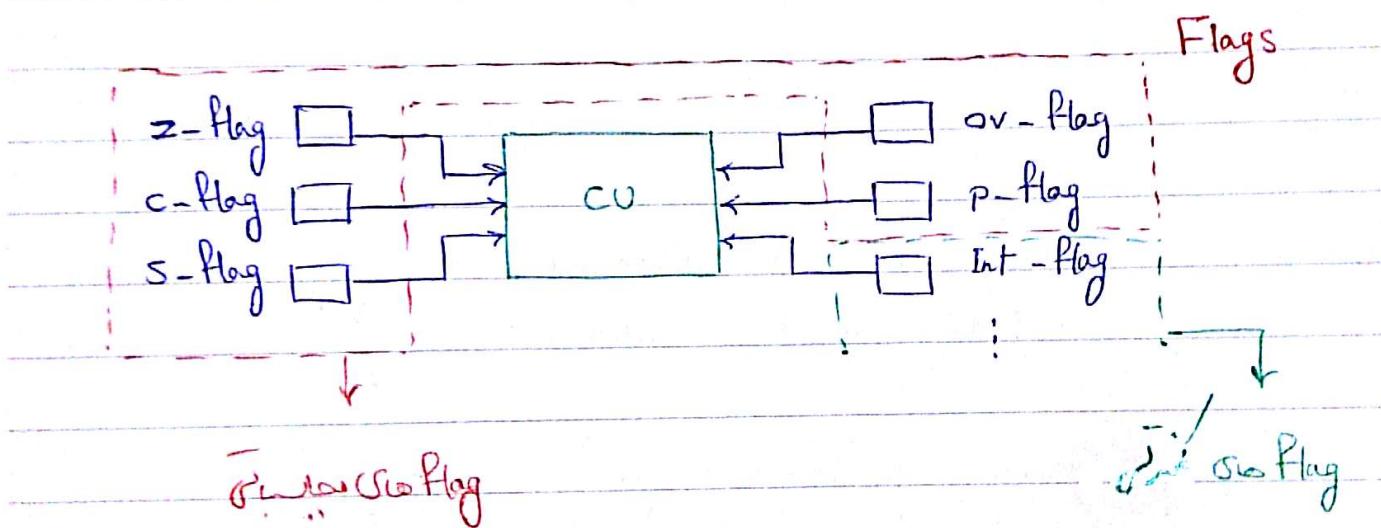
\* دستورات استاده از درسته ای:

(PUSH / POP) پرمه ۱

(CALL / RET) عنده متسق ۲

Stack را که بین این دو دستور استاده نه: Sub-Routine چیزی که CALL

Stack را که بین این دو دستور استاده نه: Sub-Routine را که RET



زنگنه زیرین: Zero-Flag  
زنگنه سینه زیرین  $\leftarrow ZF = 1$

زنگنه بیرون از زیرین: Carry-Flag  
زنگنه بیرون از زیرین  $\leftarrow CF = 1$

سیگن-فلج : Sign-Flag

سیگن-فلج مثبت است !  $Sf = 1$

سیگن-فلج منفی است !  $Sf = 0$

سیگن-فلج و اورفول-فلج : VF / Overflow-Flag

سیگن-فلج طبی !  $Vf = 1$

$01001010 \rightarrow 79$

$+ 01011100 \rightarrow 92$

$\hline 10100110 \rightarrow \ominus$

منفی  $\leftarrow$

بر تصریح رسید سیگن-فلج ( ناامان باد این آرچیتکچر ) ! صراحتاً ؟

عدد ۸ سیگن-فلج . نایابی از ۱۲۷-۱۲۸ لومینر داشته باشد !

اما حاصل ضعیف دو عدد ۷۹ و ۹۲ مرسن ۱۶۶ که از ۱۲۷ بزرگتر و تقریباً ۸

می-باشند . برای همین از بین دو عدد ممکن است در نظر گذشته باشند !

پاریتی-فلج : طبق مفهوم این فلجه از بیان زدن ارسال نماینده می-باشد . مثلاً  $10100001$  می-تواند این را مختص نمایم اطهاره از این طرف دریافت نماییم درست ، هیبت اینها را نمی-توانیم خوب نماییم !

از طریق این نمایندگی اینها را حسن نیز یا خوب نیز نمایند ( توانی و بیخی ) Even-Parity .

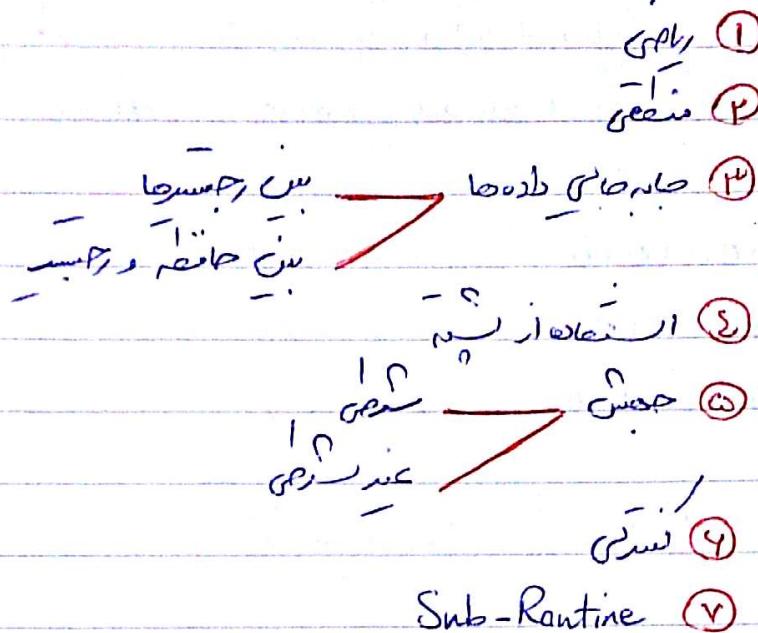
معنی تعداد ایجاد نیز بیخی ، ( توانی نیز ) Odd-Parity می-باشد .

در اینجا مفهوم نمایندگی نیز نیز است . نایابی

از این تعداد ایجاد نیز بیخی !

Flag ها در دستورات ۳۲ بیتی است \* طرد

### أنواع دستورات (عمليات)



MIPS أو سطرين من الـ 

مسح حقول این سطرين :

1. بخطه ۳۲ بیتی است (دارد دو عدد ۳۲ بیتی است).

2. ALU . سهم ، ۳۲ بیتی است.

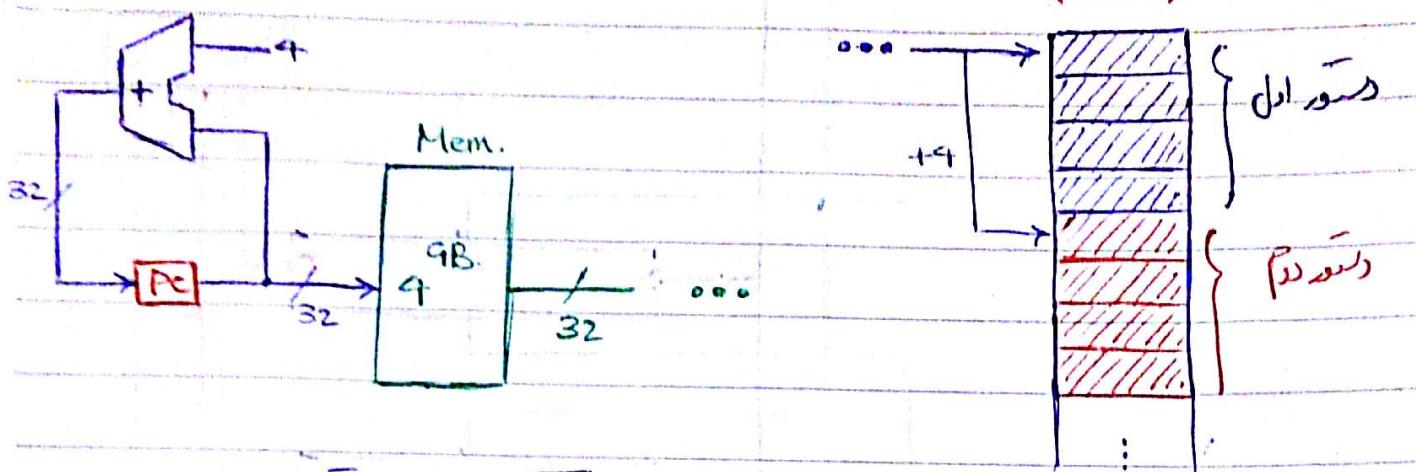
. ۳۲ بیتی است GPR +

3. ۴ Gigabyte حجم حافظه و حافظه برخانی ۴ بیتی است.

4. صحن [ ] دستورات ۳۲ بیتی است .

5. آنچه میتوان طبل اینجا میتوان ۹۸ کمل است .

مطابق (متعدد) معلم (نقطة)



جوابِ سوالات

$$4 \text{ GB} = 2^2 \times 2^{30} \text{ B} = 2^{32} \text{ B}$$

\* سُلْ بَلَ ، كُوكِيْ وَالْمِنْ دِسْوَارَ (Fetch) لَا تَشْعُنْ يَهْ !

دِسْقَرْ مُورَدْ تَقْدِيرْ تَرْسَطْ PC آدَمْ دِهْ مِنْ سُلَّدْ وَكَدْ دِسْقَرْ اَزْ حَاجَمْ كِبْرَاهِهْ خَرْفَانْ

مِنْ سُلَّدْ . خَنَّا ، PC بَلَدْ بِهِ دِسْقَرْ لَهْ اَسْدَهْ لَهْ (حَصْلَهْ هَرْ دِسْقَرْ 4 بَلَتْ

اسْتْ ، بَلَهْ دِهْنَهْ دِهْ فَخَانَهْ ! بَانَهْ حَلَّهْ رِفَعَهْ ) .

! Decode مراجع کے حوالہ میں

## : Decode

معلمون سند درسته والمعروض

ADD  $r_d$ ,  $r_{S_1}$ ,  $r_{S_2}$  ( $r_d \leftarrow r_{S_1} + r_{S_2}$ )

```

graph TD
    dest[destination] --> source1[source 1]
    source2[source 2] --> source1

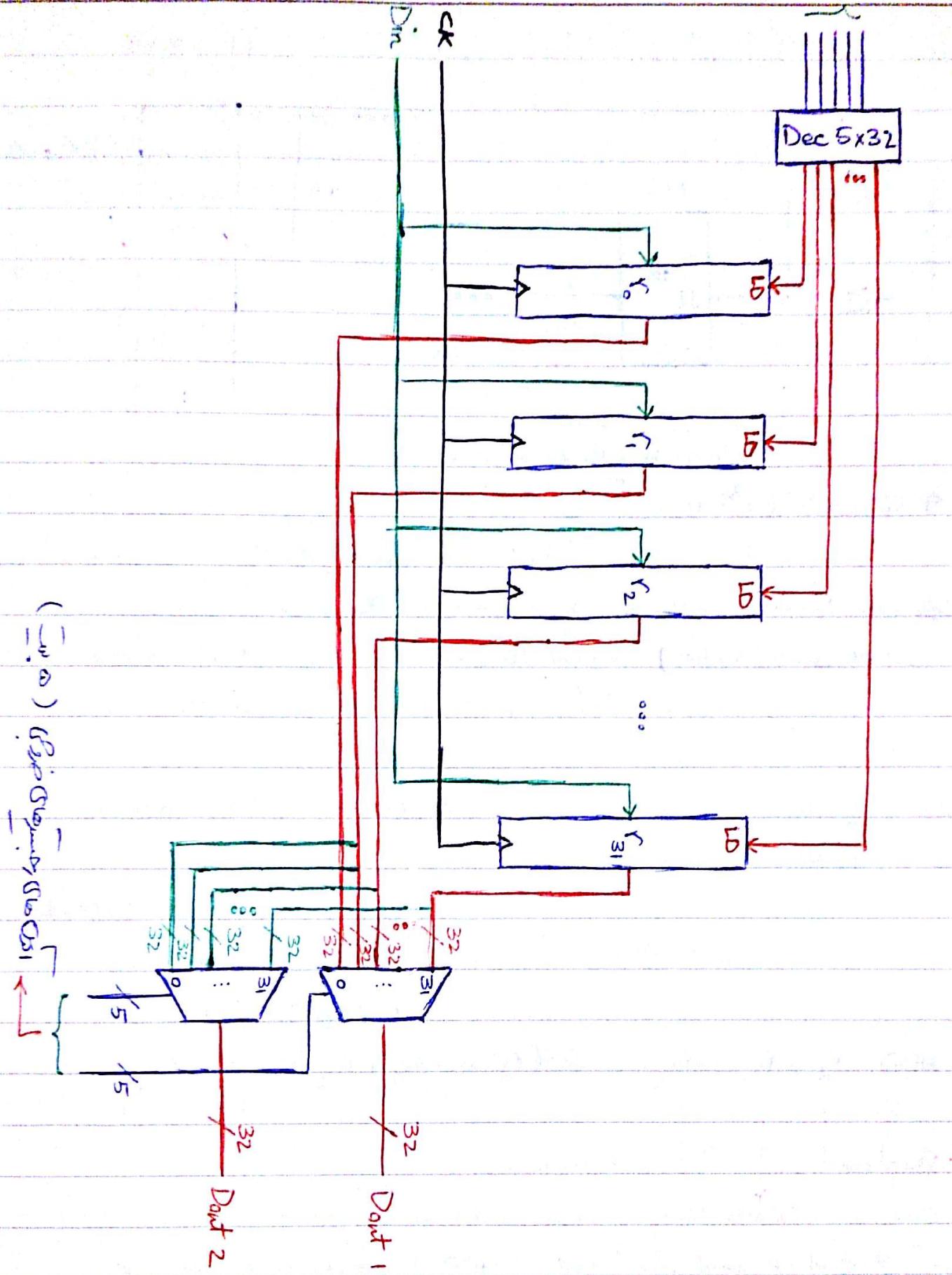
```

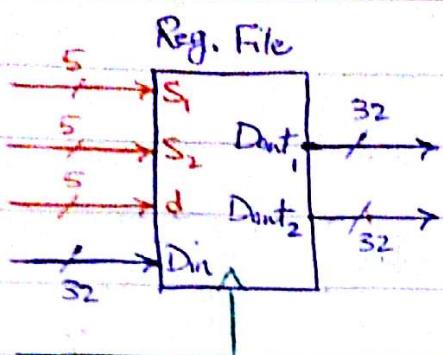
! El el uso de los índices (GPR) es una clase de índice

DATE / /

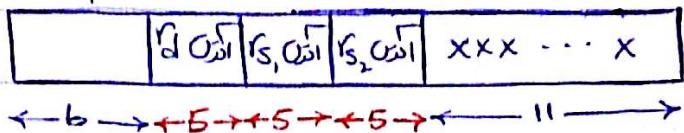
SUBJECT: 02

( $\Sigma w_i$ ) ( $S_{31}$ , ...,  $S_0$ ) ( $U_{31}$  ←)





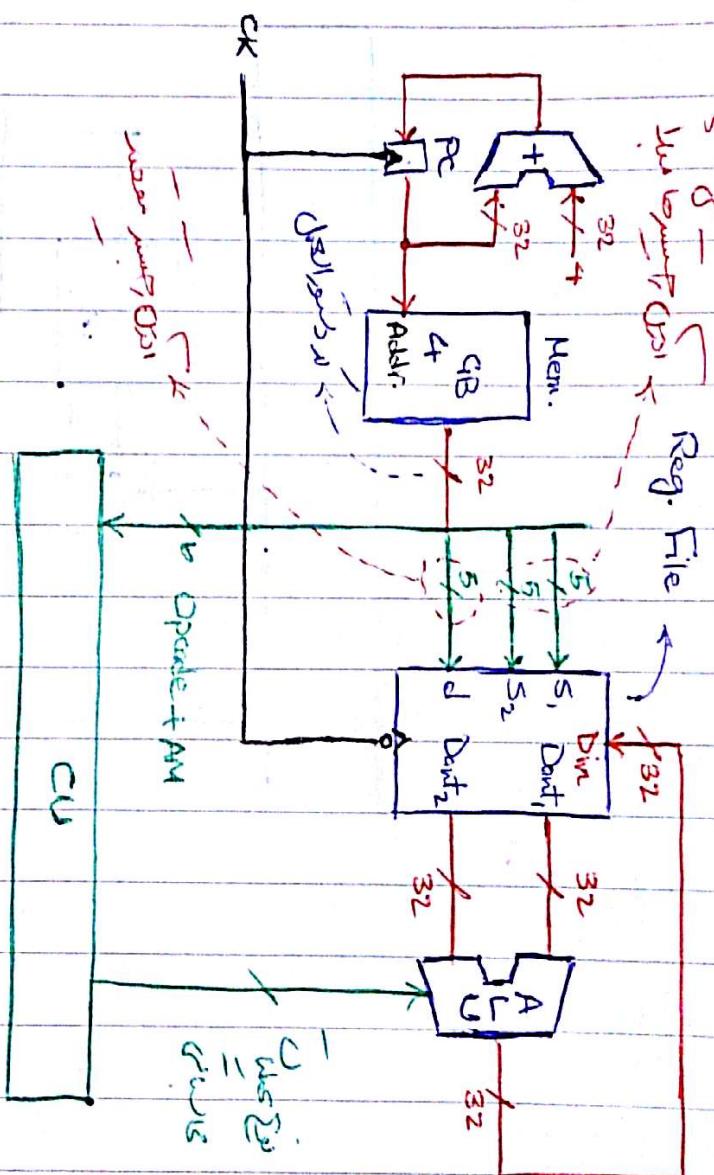
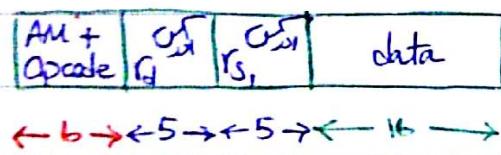
AM + Opcode

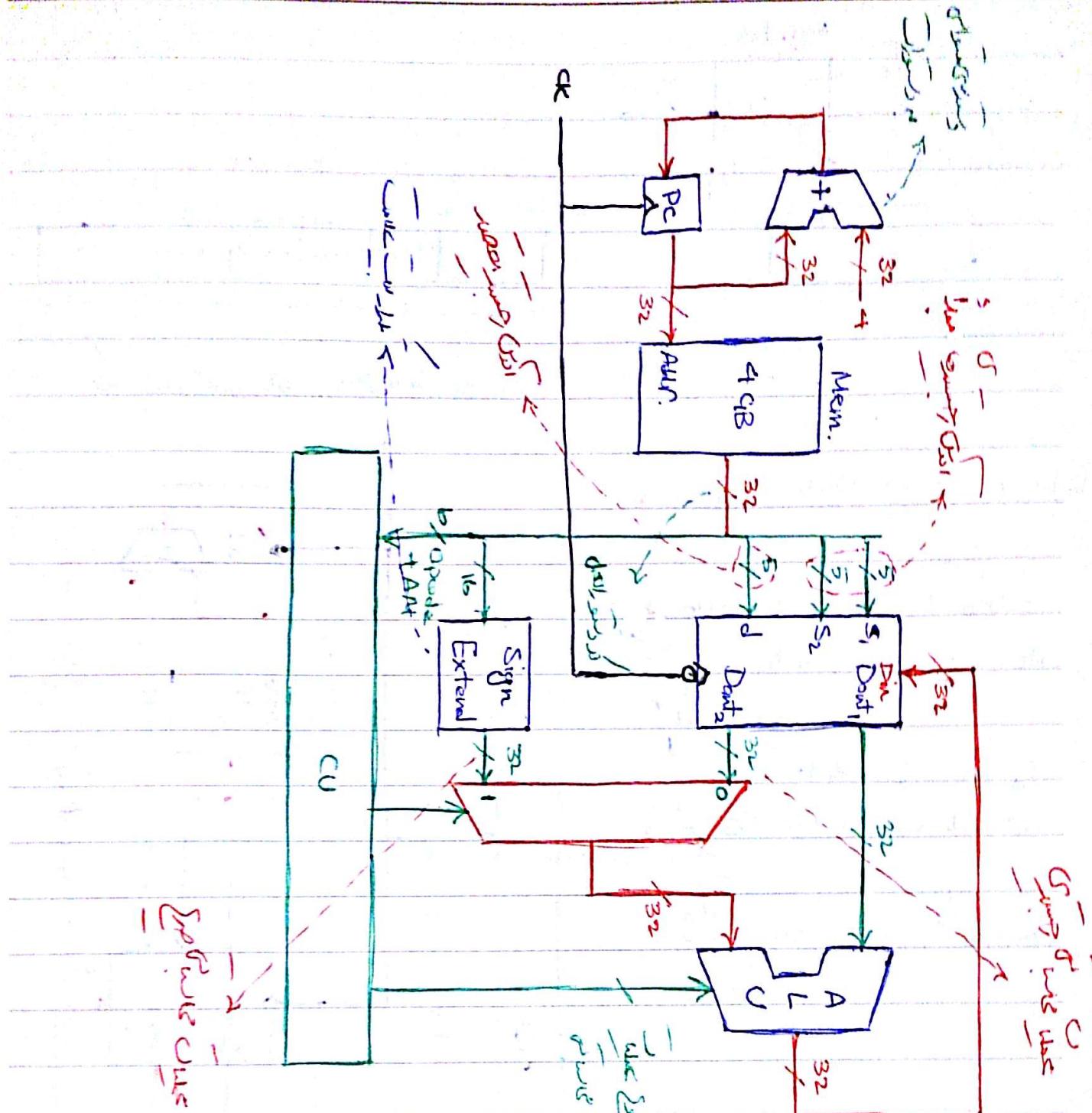


Don't Care

چیزی که در CPU، علی‌گویی نمایند

(جواب) Single Cycle - CPU \*

ADD,  $R_d$ ,  $R_s$ , data $R_d \leftarrow R_s + \text{data}$ نحوی ل پردازی از نوع  $O_1 = 0$  است \*نحوی ل پردازی از نوع  $O_1 = 1$  است \*نحوی ل پردازی از نوع  $O_1 = 1$  است \*



٩٤، ٢، ٩

(پردازش)

لود / Load (D)

سرو / Store (R)

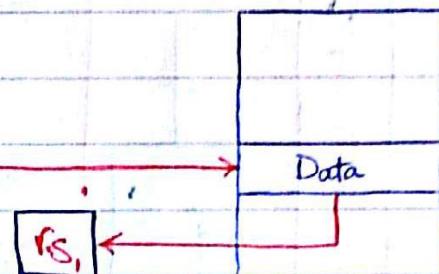
کارڈریج

LD  $r_d$ , Base( $r_s_1$ )

کارڈریج کو ارجمند کریں

Effective-Addr. = Base + Index

کارڈریج



ab(s, i)

opcode	$r_d$	$r_s_1$	Base
6	5	5	16

اُنچھے کام کرنے کا ایک طریقہ \*

! سرو جو نہ فرم سکے \*

\* RD : Read / رفتاری ارجمند

\* WR : Write / زینتیں درج کریں

Sahand

Print to PDF without this message by purchasing novaPDF (<http://www.novapdf.com/>)

$l_0 \rightarrow R_1 \leftarrow R_{S_2} *$

! out reg

User Base Addr. \*

! 16 bit

! 32 bit extend

Reg. Ext. \*

Out reg ALU

! 32 bit

Reg. DR \*

Data Mem. DR

! DR RD DR

Reg. DR \*

Reg. File

ALU Reg. DR

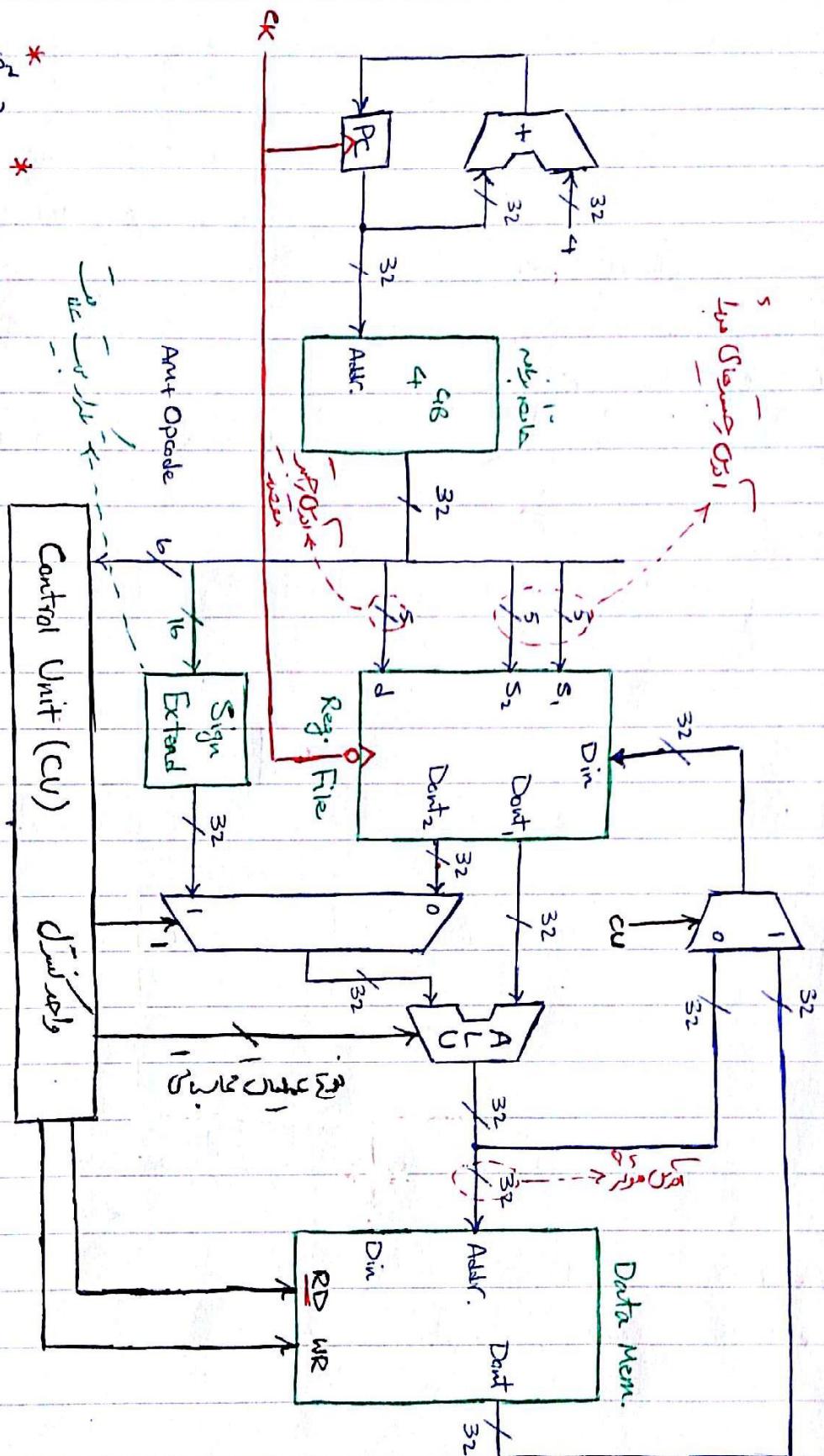
! 32 bit

Reg. DR

! 32 bit

Data Mem. DR

! DR RD



فان سر دسته و مسی سر

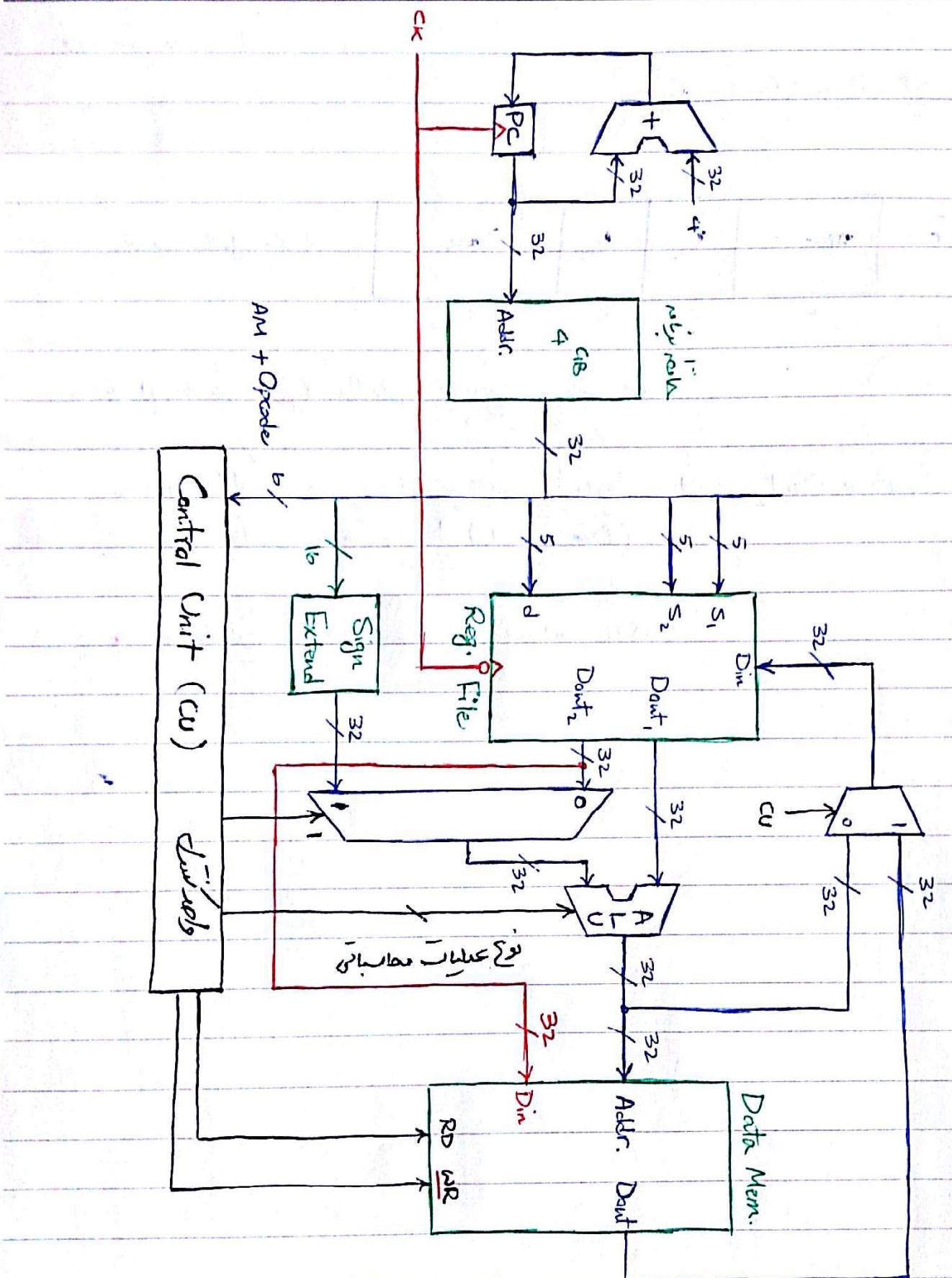
ST Base( $r_{S_1}$ ),  $r_{S_2}$

العنوان	opcode	$r_{S_1}$	$r_{S_2}$	Base	لدي دسته الامر
	b	5	5	16	

\* این جا داده باید  $r_1$  باشد . پس از آن  $r_2$  باشد

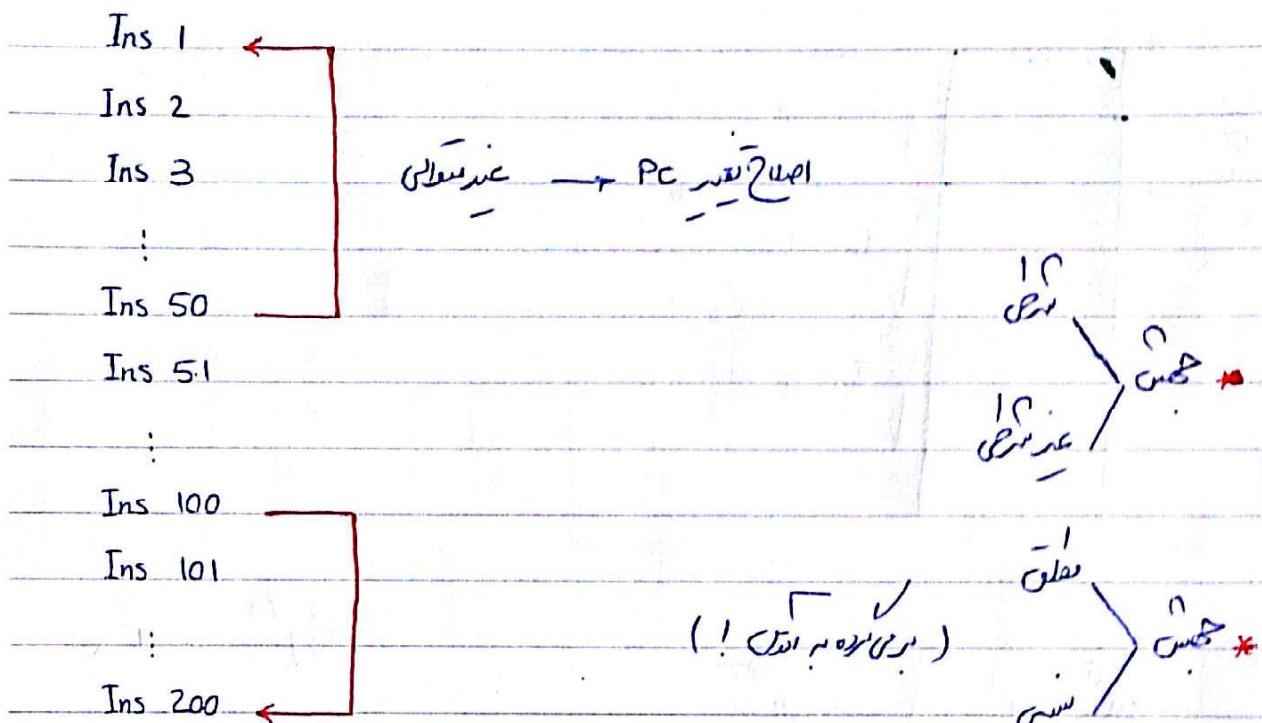
Job or Data  $\rightarrow$  کار یا داده است . این داده را در حافظه ذخیره کنید .  
Din  $\rightarrow$  این داده را از حافظه خوانید !

\*  $r_1$  و  $r_2$  این اندیشه خارجیست ! باید دسته داشته باشند



(مسئلہ چھپ)

مسئلہ چھپ میں سے اسی میں سے تینی کو اجرائی برام از حدات میں خارج نہ ہو۔ مل  
حکم ہا و نیطا ہا \*



مسئلہ چھپ میں اسی طبقہ کو اجرائی برام از حدات میں خارج نہ ہو ! \*

(! اور bneq flag اور IP کو حفظ کرنا ہے ) = to flag, to bneq, مدد ہے \*

درستہ نہیں تو پیچھے چشم :

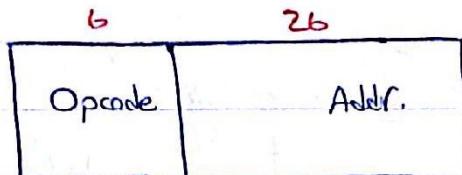
, JMP Addr

, bneq RS<sub>1</sub>, RS<sub>2</sub>, Rel-Addr.

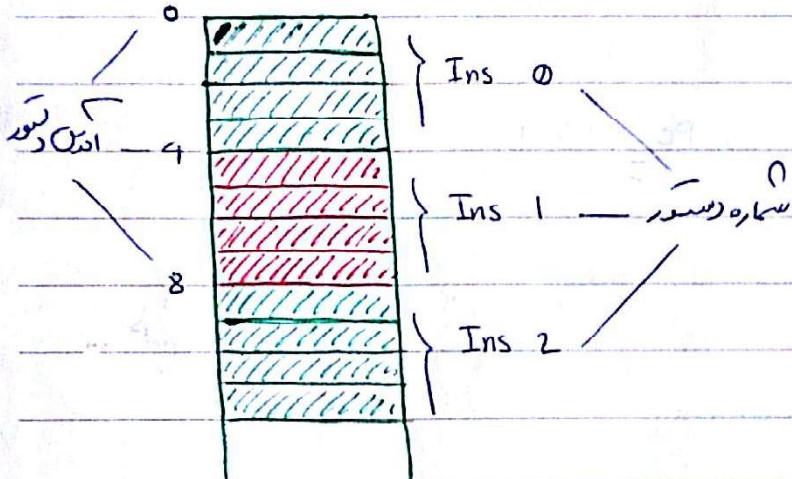
امتحان - IP - مذکورہ و مذکورہ اولیا !

JMP Addr.

نحوه دستور والدرس سمت به معرفت درین بروگاه  
در دستوراتی که معرفت نزد است.



ادریس طبقه ای دستور  $\Rightarrow$  دستور دستور



\* با قوی بی سعی باید

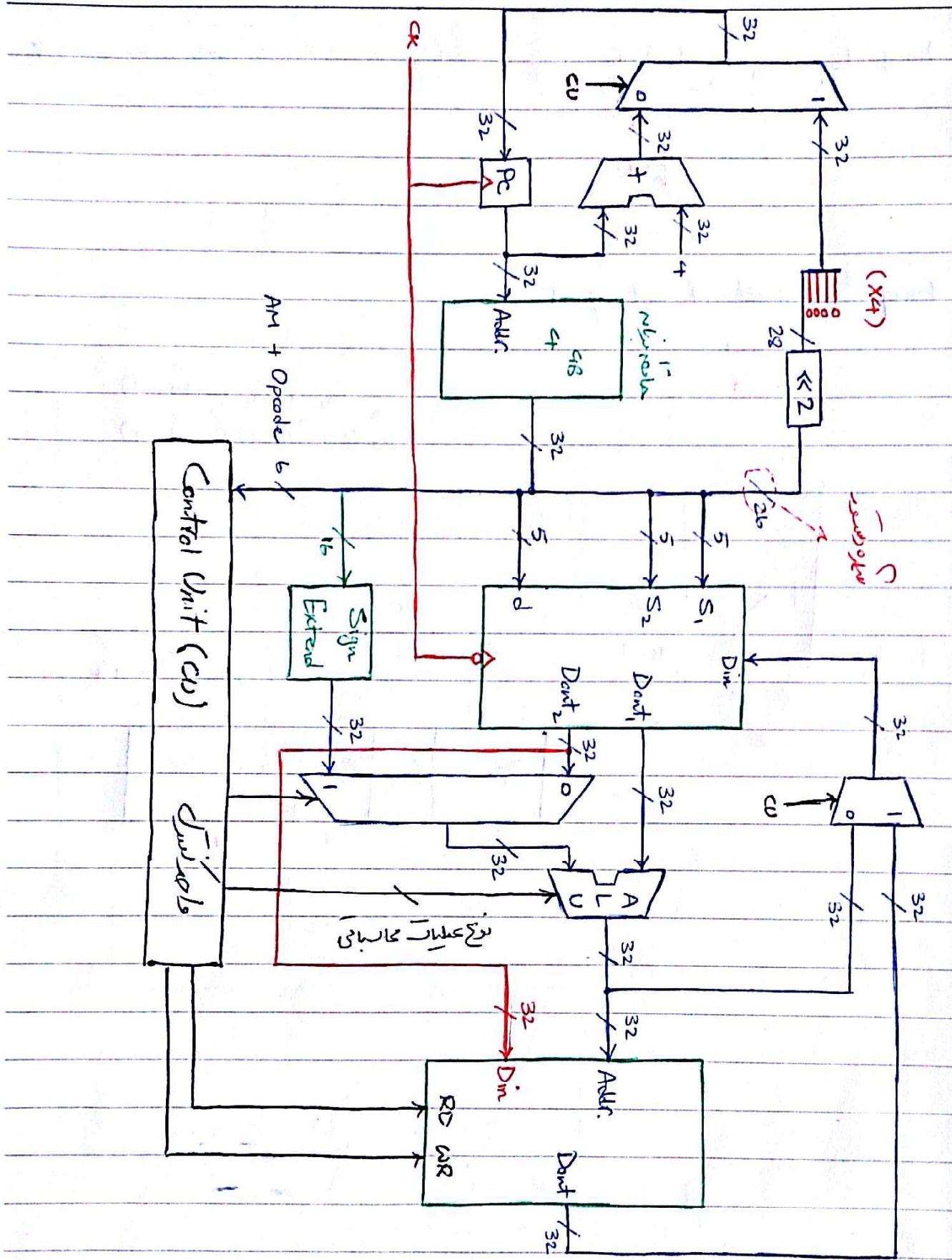
$$\text{ادریس دستور} \times 4 = \text{سیاره دستور}$$

\* دری میکنید، هر جا سیاره سمعت طلب، مرتبط نمی‌نمایی ! ( > 2 )

\* سیاره سمعت داشته باشد !

\* در دستوراتی که معرفت ندارند، اول به ترتیب رنگ می‌برند و بعد دستوراتی که معرفت داشت باشند احتمال داشتند

\* در دستوراتی که معرفت داشتند، دستور حفظی ( حافظه ) !



DATE / /

SUBJECT: ۹۸

bneq  $r_{S_1}$ ,  $r_{S_2}$ , Rel-Addr.

مسار دستور

$\downarrow \times 4$

ادخل دستور

bneq : branch if not equal

اسعاد

$r_{S_1}, r_{S_2}$  مطابق ۱

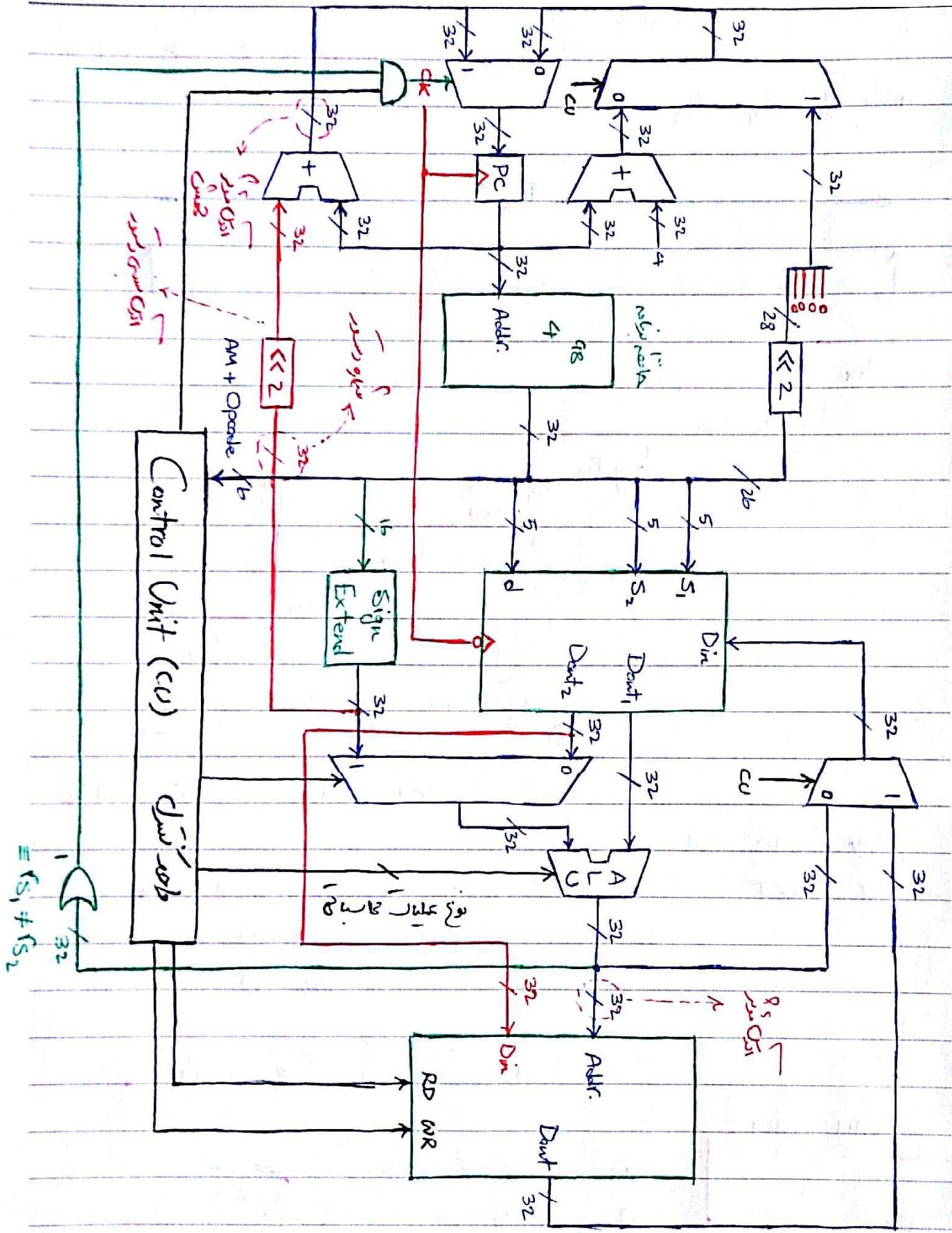
غير مطابق ۲

bneq

اسعاد

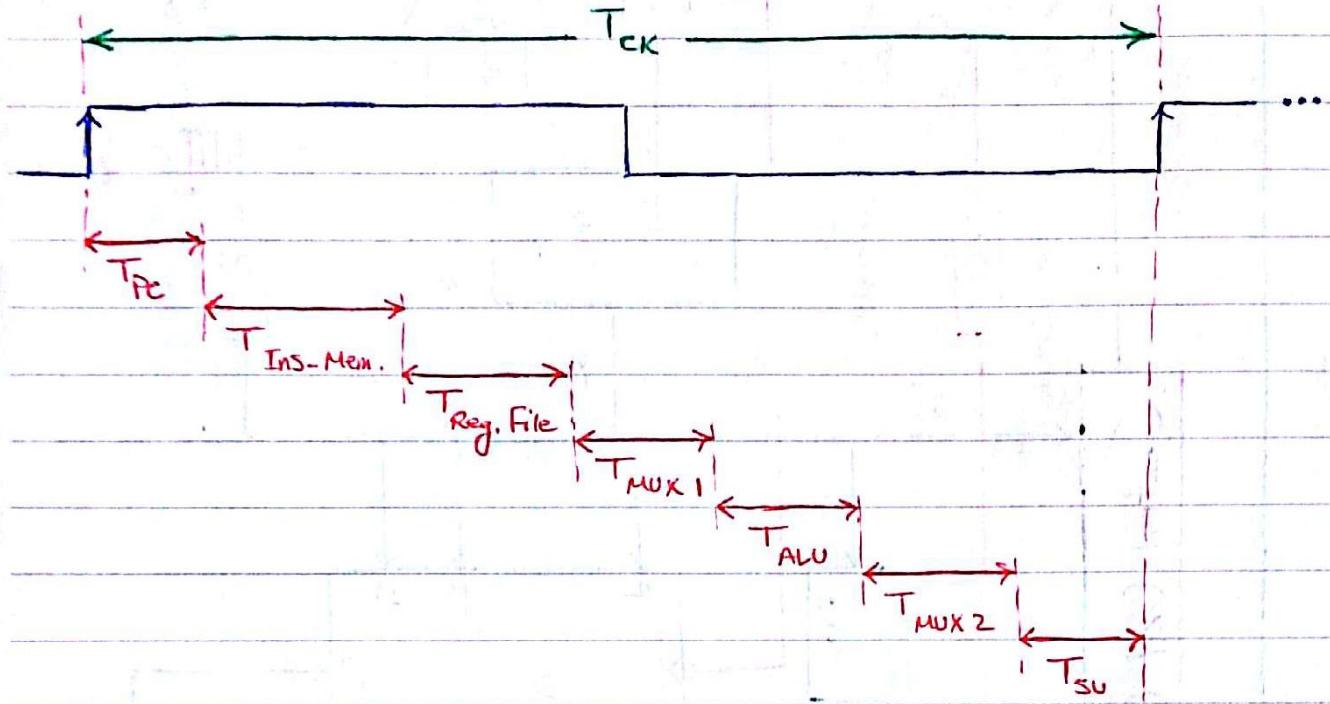
برهان دستور بحسب ترتیب \*

Opcode	$r_{S_1}$	$r_{S_2}$	Rel-Addr.
6	5	5	16



معنی ساختاری: ۱) نی

ADD D - R<sub>S1</sub>, R<sub>S2</sub>



لهم  $\downarrow$  Setup-Time  $\downarrow$

RET, CALL لئوس؛ ۱)

فرازی CALL Sub

- لئوس RET

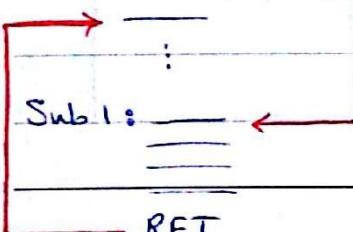
: (Sub-Routine) فرایند است که \*

لئوس ۱)

لئوس ۲)

! لئوس با لئوس ۲ : CALL .

CALL Sub 1



۱) Save return address.

۲) Jump to Sub.

} CALL

$$\text{Speed up} = \frac{N \times \sum_{i=1}^K T_{\text{stage},i}}{(N+K-1)(\text{Max } T_{\text{stage},i} + T_{\text{ov}})}$$

لطفاً من اسرعه دستگاه، میتواند بروزگیری کند  
میتواند بروزگیری کند این نتیجه میتواند باشد  
نه!

$$\text{Speed up} = \frac{N_1 \times \sum_{i=1}^3 T_{\text{stage},i} + N_2 \times \sum_{i=1}^4 T_{\text{stage},i} + N_3 \times \sum_{i=1}^5 T_{\text{stage},i} + \dots}{(N+K-1)(\text{Max } T_{\text{stage},i} + T_{\text{ov}})}$$

۹۱، ۹۲، ۹۳

### Pipeline Circuits

① خالص طبق

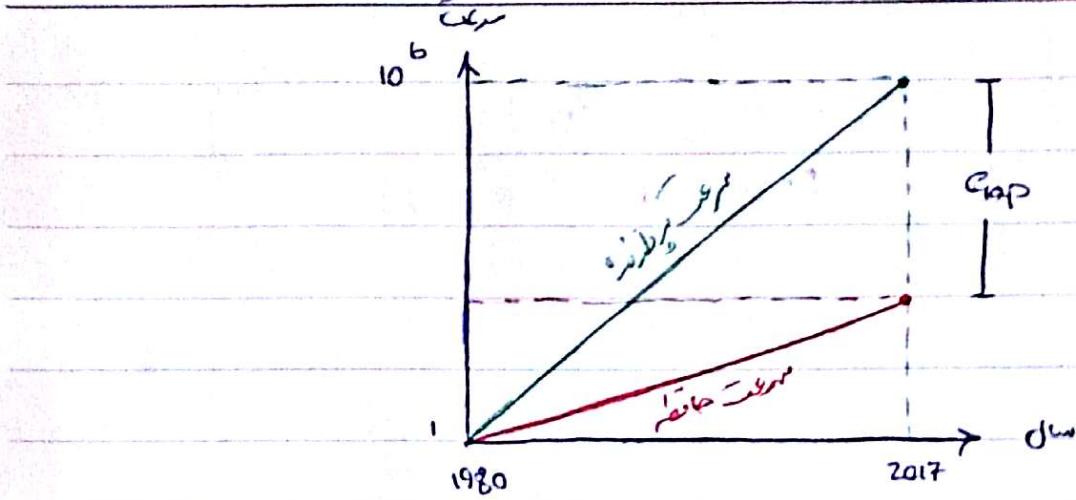
② خالص سایری

③ خالص اول

### Cache Memory / حافظه های

ساده کنیم که حافظه سیستم هست. میتوانیم مفعلاً میتوانیم این را  
است و درستی به حافظه های زمان برآورده است.

کافیست که حافظه سیستم طبق، مبتنی بر مکانیسمی باشد که میتواند این را  
نمایش دهد!



\* حد ناچاری سرعت میں بیانیہ وحاظم چیز؟

- ① اولین سرعت بروزگفتہ حاصل نہیں ہے۔
- ② حجم حاضر حال مدد استفادہ اولین یافہ، دستیکر وہ سرعت ان حاکم ہے۔

$$\text{اولین حجم حاضر} = \text{اولین سرعت}$$

\* سہ عمل در پیغام ایک مرتبہ Fetch, Execute, Decode, Fetch،  
کو دستور از حاضری برنامہ خاتم ہے لہد، لیکن ہر دستور نیاز ہے جو حاضر طور

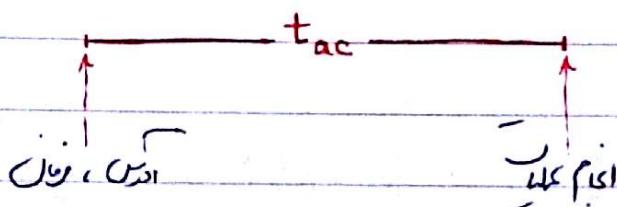
\* سرعت بروزگفتہ زیاد، اور سرعت حاضر نہ! جیل بروزگفتہ می خواهد از حاضر استفادہ نہ، میں کبھی  
کو سرعت اور با حاضر کیونہ! و اندر میں حال ما بری اولین سرعت بروزگفتہ برمایوں ہوں!

\* کو کوئی اپن سطح نہیں، اور سرعت حاضر خوب خوب ہے، جیسے باریم ہے۔ لذتمن، اور وہ  
برنامہ ہی بسیں تولید ہوئے ہے بے حجم حاضری زیادی نیاز طور، این ہے شاخصہ! باریم ہے؟

پیغامی سیار محض و صدر طور!

## Locality / مکانی

اگر دستوری کامپیوٹر کے یادگار میں موجود ہے تو اس کا دستوری زمانہ کم ہے۔ اس کا دستوری زمانہ اس کے دستوری سرعت کے مطابق کم ہے۔



اگر دستوری کامپیوٹر کے یادگار میں موجود ہے تو اس کا دستوری زمانہ کم ہے۔

### Locality / مکانی

	$i_0 : t_0$	$t_0 + \Delta t$	Temporal Locality
اگر دستوری کامپیوٹر کے یادگار میں موجود ہے تو اس کا دستوری زمانہ کم ہے۔	$ob : Data(K)$	$Data(K)$	( $i_0 \rightarrow i_1$ )
	$i_0 : t_0$	$t_0 + \Delta t$	Spatial Locality
اگر دستوری کامپیوٹر کے یادگار میں موجود ہے تو اس کا دستوری زمانہ کم ہے۔	$ob : Data(K)$	$Data(K \pm \delta)$	( $i_0 \rightarrow i_1 \pm \delta$ )

اگر دستوری کامپیوٹر کے یادگار میں موجود ہے تو اس کا دستوری زمانہ کم ہے۔

اگر دستوری کامپیوٹر کے یادگار میں موجود ہے تو اس کا دستوری زمانہ کم ہے۔

مسن  $t_0 + \Delta t$  بـ  $\sigma_{Data}(K \pm \delta)$  مابد  $Data(K)$  نیاز نایم.

من مسنه اینچنان طرس و من خواست دری بخوشت . ساعت  $\pm$  پنج فعل اول او میرود من خواست بـ  $\sigma_{Data}(K \pm \delta)$  حمل فعل و دارید من خواست ! ساعت  $\pm$  هشت تا هشت و نیم ! ساعت  $\pm$  نیزه سراغ به دری داشت ، آن دلیل در میان چند فعلی نداشت .

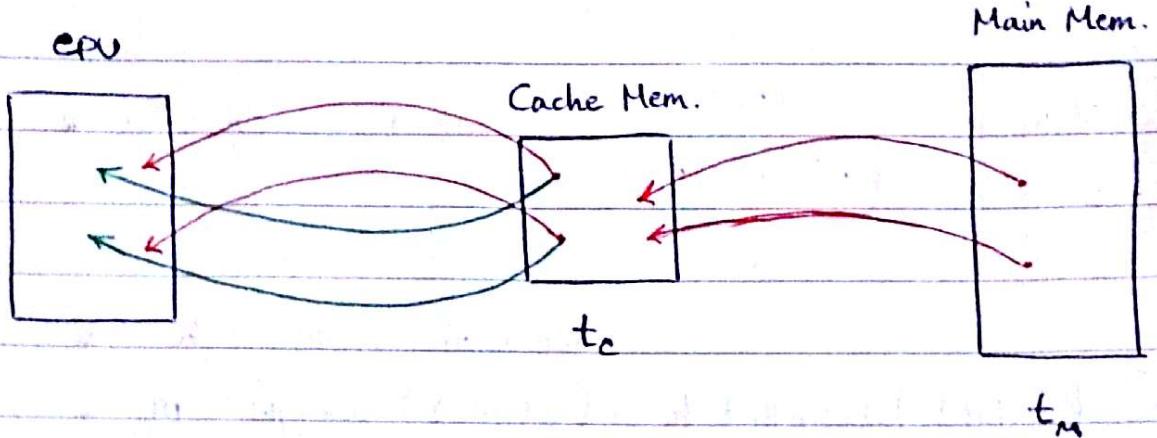
اله در زبان  $t_0$  حاسیت طوف و زبانهای او احتمال نیست ، انسی بعد هم چنین زبان نداشته است ، خود در زبانهای دیگر او خواسته ! چنانچه مبنی  $Data$  ها او استاده نیست  $\sigma_{Data}$  ها کیا فواید را !

آن اینکه زن !

اله مسنه مکاریه همچنان دل رو اورد بخوشت (!) ، باید طوف نهاده شد ! لعن اله در زبان  $t_0$  طوف اون دلها را در من خواستم ، در زبان  $t_0 + \Delta t$  هم هر دلها را من خواستم نه دلها را داشتم !

جذب و جذب  $t_0$  ، ما دله به جاهش بین ساز نایم . به عین حال ، ما اله می خواهیم سه کوئیم ، اون مسنه طوف سه کوئی او خواهیم ! اون سه کوئی در زمانه داریم و من خواهیم . اله سه کوئی خواهیم که نایمه بودیم ، صحب میخواهیم خواهیم ! در مفعع به جلد میست زمانه میخواهیم .

حاله که خواهیم  $t_0$  خود طوف ، ساعت  $\sigma_{Data}$  همچنان دل رو دریغ من خواهیم .



$t_c \ll t_M$  \*

برمجه Cache دا بيمه خاصه داره Data را جيئونه اگر نهاده باشند. ساريان تعداد Cache برومطيح و استفاده هوش است. Main Mem. يخواه هر دفعه داده را در Cache را ذخیره و مركب باشند.

جهاز Cache را مطالعه داشتم باشند. مطالعه باشند جهاز Cache خاص است. خواهد باشند مواد !

برای عرض مفهوم نظام هادر داشت که به بروز خواهد شد ! اما دستورات در آن زمان نداشتند !

### Hit-Rate

دستور / موضع : درج / موضع : Hit \*

نمان : درج / موضع : Miss \*

$$\text{Hit-Rate} = \frac{\# \text{hit}}{\# \text{hit} + \# \text{miss}} = P(\text{hit}) : \text{نیزه را داشتند} *$$

$$\text{Miss-Rate} = \frac{\# \text{miss}}{\# \text{hit} + \# \text{miss}} = P(\text{miss}) \quad \text{نیزه سان} *$$

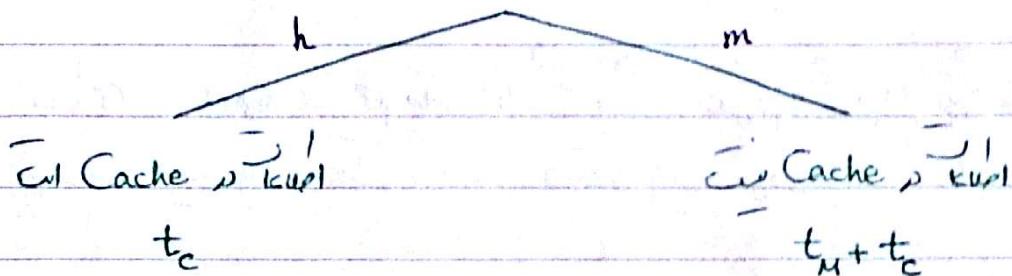
(m)

$$h = 1 - m, \quad m = 1 - h \quad *$$

نیزه (P(hit)) یعنی  $h \rightarrow (P(\text{miss}))$  یعنی  $m$  \*

مقدار  $h$  و  $m$  را چگونه حساب کنیم؟

Data  $\rightarrow$  نیزه



نیزه اینجا کاش و مموری هست

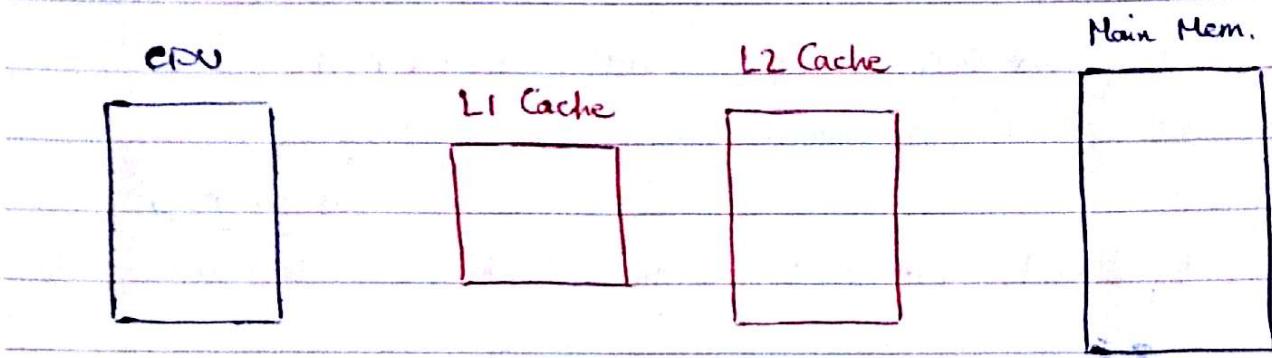
$$t_{av} = h \times t_c + m \times (t_m + t_c) = ht_c + mt_m + mt_c$$

$$= (h+m)t_c + mt_m = t_c + mt_m$$

?  $t_{av}$ ,  $t_c$ ,  $t_m$ ,  $h = 0.95$ ,  $t_c = 2^{\text{ns}}$ ,  $t_m = 100^{\text{ns}}$

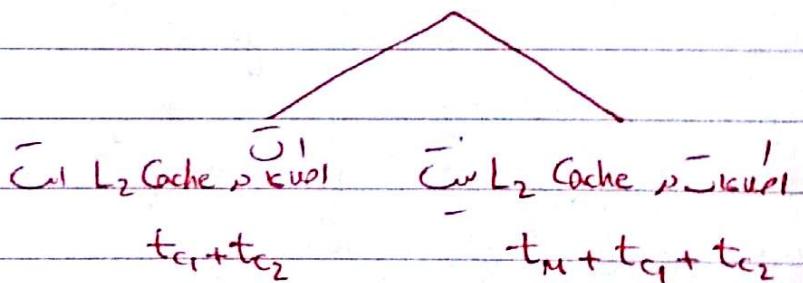
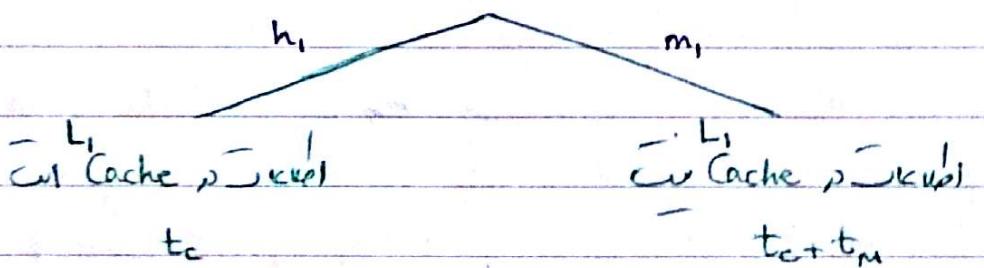
$$t_{av} = 2 + 0.05 \times 100 = 7$$

نحوهی میخواهیم که این دو روش را برای اینجا در نظر بگیریم  
 (Memory Hierarchy) پیشنهاد میکنیم



L : Level / Layer

Data requests



$$t_{av} = h_1 t_c + m_1 (h_2 (t_{c_1} + t_{c_2}) + m_2 (t_{c_1} + t_{c_2} + t_m))$$

$$\Rightarrow t_{av} = t_{c_1} + m_1 t_{c_2} + m_1 m_2 t_m$$

$h_{c_2} = 0.95$ ,  $t_{c_2} = 10$  ns,  $h_{c_1} = 0.9$ ,  $t_{c_1} = 2$  ns,  $t_M = 100$  ns.  $t_{av}$  ?  
?  $t_{av}$   $\leq t_{ar}$ .  $t_{av}$

$$t_{av} = 2 + 0.1 \times 10 + 0.1 \times 0.05 \times 100 = 2 + 1 + 0.5 = 3.5$$

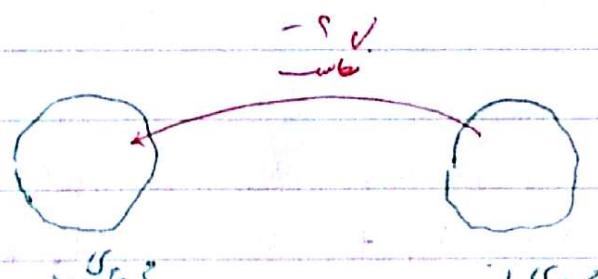
:  $t_{av} \leq t_{ar}$  \*

$$t_{av} = t_{c_1} + m_1 t_{c_2} + m_1 m_2 t_{c_3} + m_1 m_2 m_3 t_M$$

ا) (جواب) ! پردازش اثیری، تا CPU و Cache  
(! نهایی)

Cache جواب

: Mapping / - QV



(جواب)

(جواب)

Cache جواب؟ اگر کامپیوٹر میں چند ٹائپیں دستیاب ہیں۔  
اگر کامپیوٹر میں چند ٹائپیں دستیاب ہیں۔

- 0 0

ابراج

1. Fully Associative Mapping ①

## Cache / Direct Mapping

(جایگزینی) (Associative - مترابه) / Set Associative Mapping ⑭

↪ Global Substitution : Fully Associative Mapping ①

للسیم من خارج منزه ! مثلاً ، فرض کنید که صندوق خالی که نظر طاری خواهد داشت

پھر ہر طبقہ دلکش مکوار ہر روزہ سسی ۳۳۳

جع کرداش و بندانه!

1. 0) regridding Cache Grids, which can be all kinds \*

وَالْمُؤْمِنُونَ لِلّٰهِ وَرَبِّهِمْ وَلَا يُشَرِّكُونَ بِهِ (١٠٢)

سُؤال : ما هي Data ميزة ؟ (Cache ، RAM ، ROM ، Hard Disk ، Floppy) .

CPU! Cache لغایتی داشته باشد و آن را در آنها نمایش دهند.

لسم امیس اور می خار؟ ہنر آئی 89251 لو۔ CPU از ۱۰۰ بھنڑا ان

طهائی نہ در آئیں 189 طبقہ Cache صست، حملن دادیں کی درز نہ آئیں

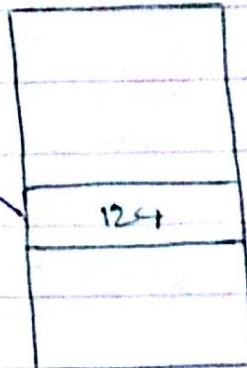
15 tel rep 89251

Tag für persönliche Werte, Cache-Jobs Schw. w:

! Data Class C1, n' Addr.

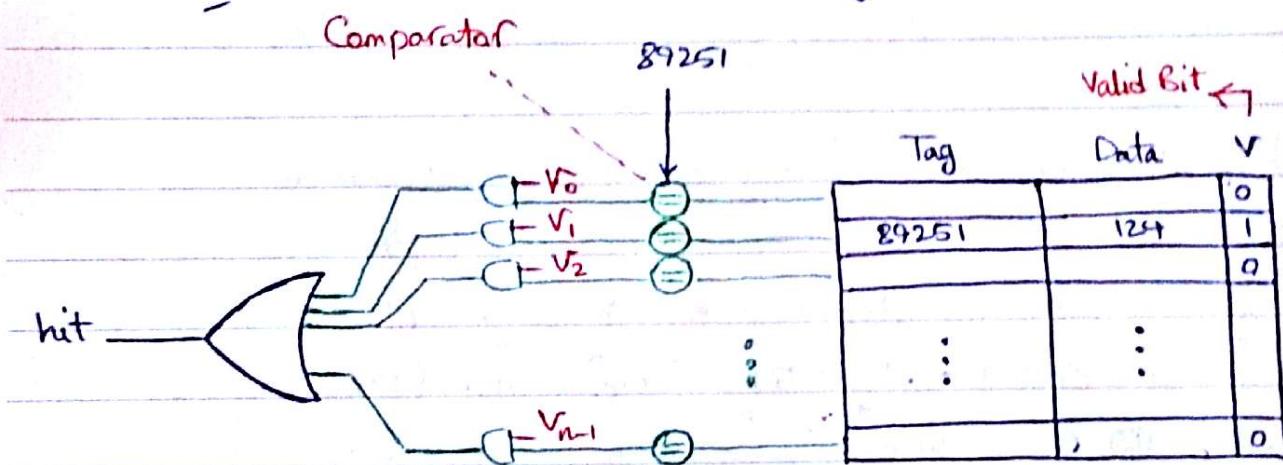
(: میں کوئی سرسری Cache چھوڑے

Tag Addr.	Data
89251	124



. 89251

\* مقدار داشته است (Comparitor) و در Cache (نیز در CPU) نیز ذخیره شده است. با اینکه CPU دخواست کرده بخواهد آدرس داده Tag را با آدرس داده داشت کرده بخواهد



و Data ایک جگہ کا وہ جگہ ہے جو کافی محدود طبقہ کا ہے اور اس کا Locality محدود نہیں۔

ایجاد سرویس و دسترسی مخصوص! اینجا چی  
درست! طبق خارج از CPU در پارس هست! از خارج مخصوص  
ایجاد! اینجا بوده که این دسترسی ایجاد!

کامپیوچر میگیرد. اینجا Cache را برای CPU، و  
کامپیوچر کامپیوچر کامپیوچر کامپیوچر کامپیوچر کامپیوچر  
نیازی نیست! Cache job، پس از برداشتم Block

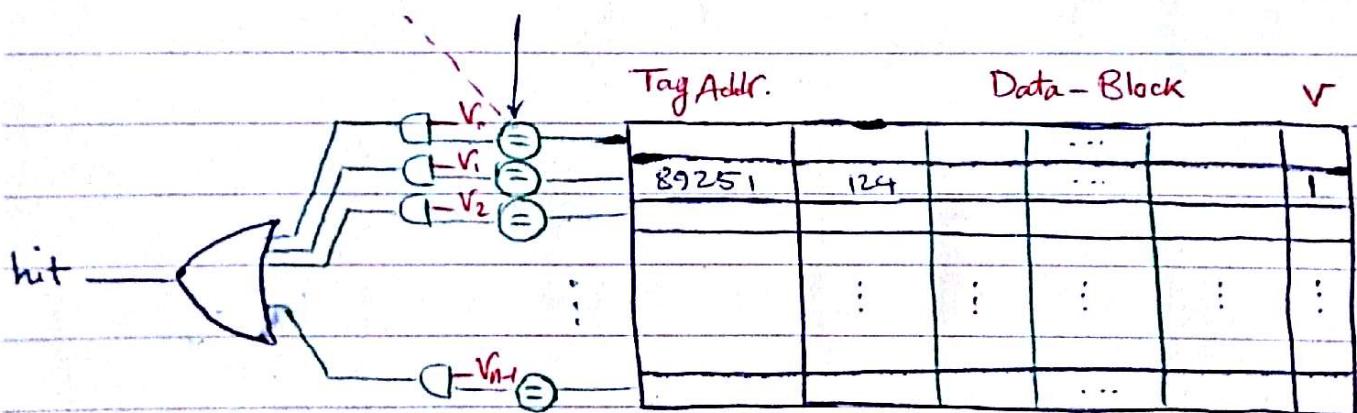
برای ۱۴ باره! این خوب است! ممکن است! Block سایز \*

\* ... ل ۱۴ ل

! Data is Block پر، Data در

Block سایز پر Tag Address \*

Comparator 89251



٩٩، ٢، ١٥

مکانیزم های انتخاب ۳۲ بتا در پردازش اینستاکت!

O/I

B0

B1

B2

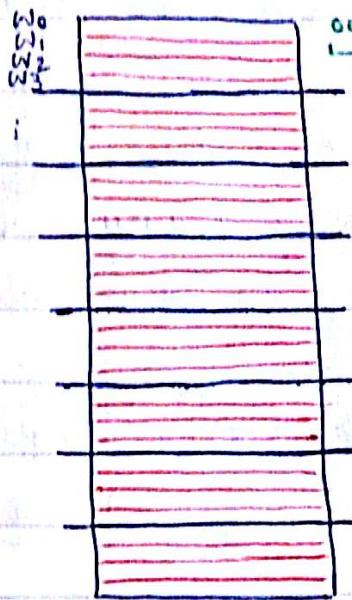
B3

B4

B5

B6

B7

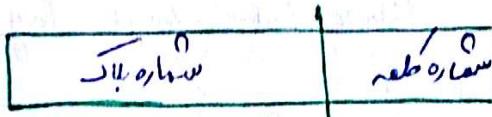


آدرس صفحه

آدرس رکور

آدرس صفحه  
آدرس رکور

پردازش اینستاکت، اینستاکت میتواند  
پردازش اینستاکت نباشد، اینستاکت میتواند نباشد



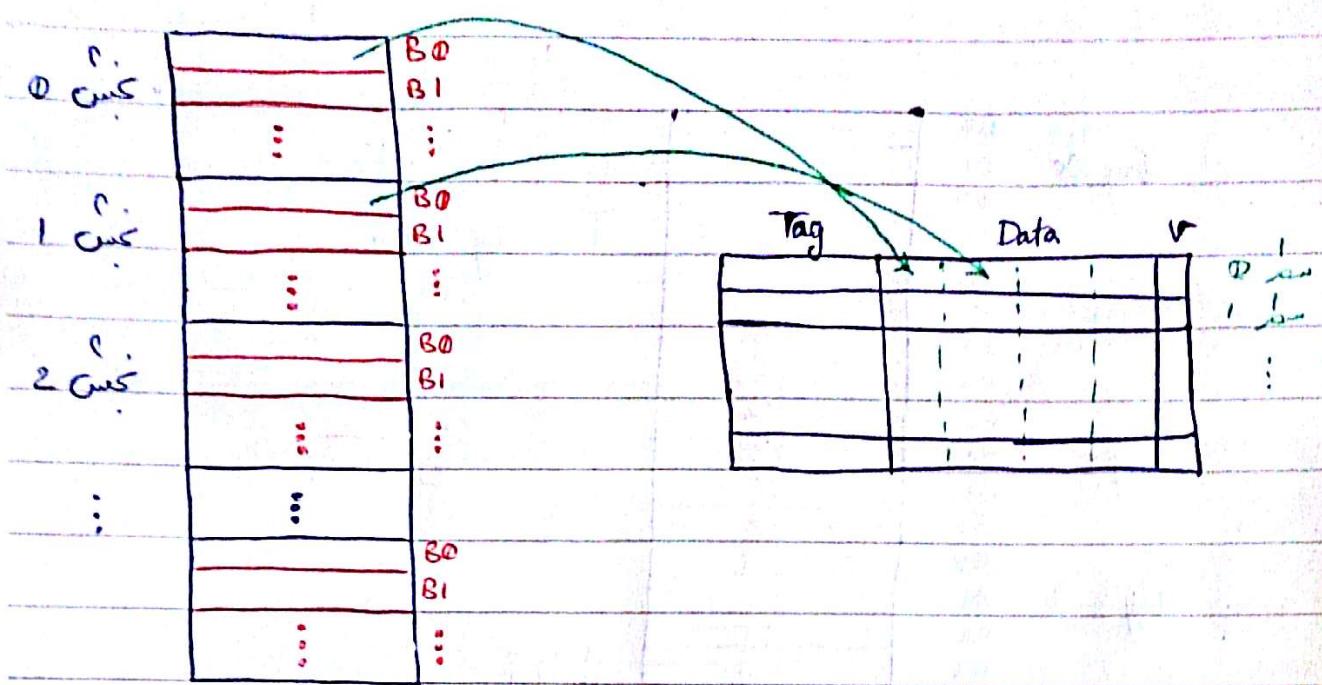
log Block-Size  
 $\frac{b}{2}$

اینکه حفظی در چندین درجاتی باشد! سلسله  
این افراد سهاره طبقه بندی کنند: بقایی در مکان و کاربرد!  
اینکه در چندین مسیر میگذرد کارهای طبقه بندی و سهاره اینکه مادر دارد اینجا!

اینکه در چندین مسیر میگذرد کارهای طبقه بندی و سهاره اینکه مادر دارد اینجا!

اینکه در چندین مسیر میگذرد کارهای طبقه بندی و سهاره اینکه مادر دارد اینجا!  
اینکه در چندین مسیر میگذرد کارهای طبقه بندی و سهاره اینکه مادر دارد اینجا!  
اینکه در چندین مسیر میگذرد کارهای طبقه بندی و سهاره اینکه مادر دارد اینجا!

جهاز محوّل آن باید اور در حافظه Cache پیشگیری شود. مدارس نسبت دینی همچنین



• Cache  $\rightarrow$  Data Storage  $\rightarrow$  Processor  $\rightarrow$  Memory  $\rightarrow$  Processor  $\rightarrow$  Cache

و سفر حافظ (خان) !

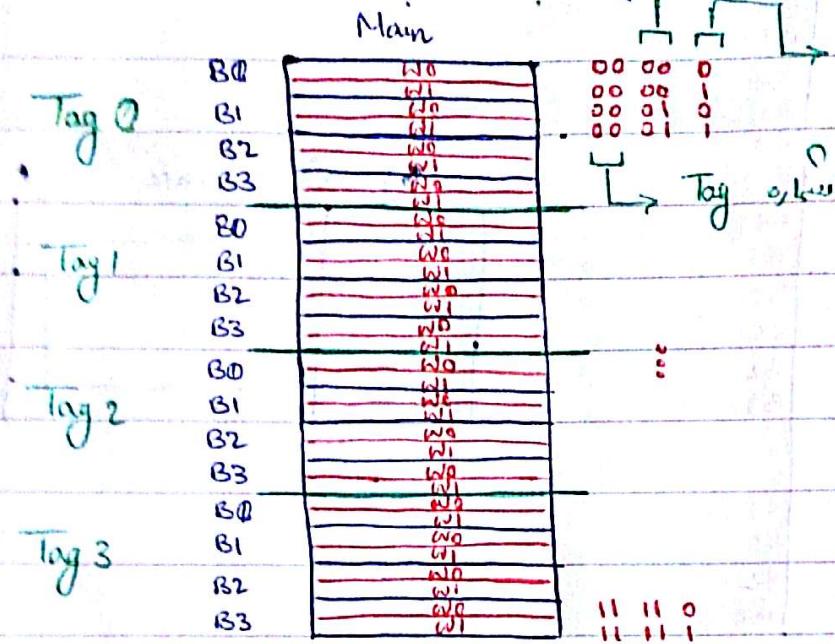
وَمِنْهُمْ مَنْ يَرْجُوا أَنْ يُبَرَّأَ مِنْ ذَنبٍ فَلَا يُبَرَّأُ وَمَنْ يَرْجُوا أَنْ يُعَذَّبَ فَلَا يُعَذَّبُ وَلَكُمْ دِيْنُكُمْ وَلَنَا دِيْنُنَا

\* اگر داده‌ها در مجموعه‌ی Data باشند و کواریانس میان داده‌ها مثبت باشد، آن‌ها را متمایز نمی‌نامیم.

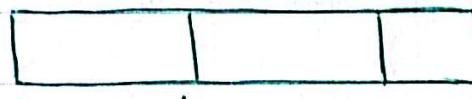
مِنْ سَارَهُ كِبِيرٌ اَوْ Tag \*  
كِبِيرٌ ( "كِبِيرٌ" اَوْ مِنْ سَارَهُ حَاجَرٌ ) .

(!  $\rightarrow$  Tag)

Suppose, we have 8 blocks per cache line and each line has 32 words.



:  $\frac{1}{\text{Block-Size}} \times \frac{1}{\text{Word-Size}}$



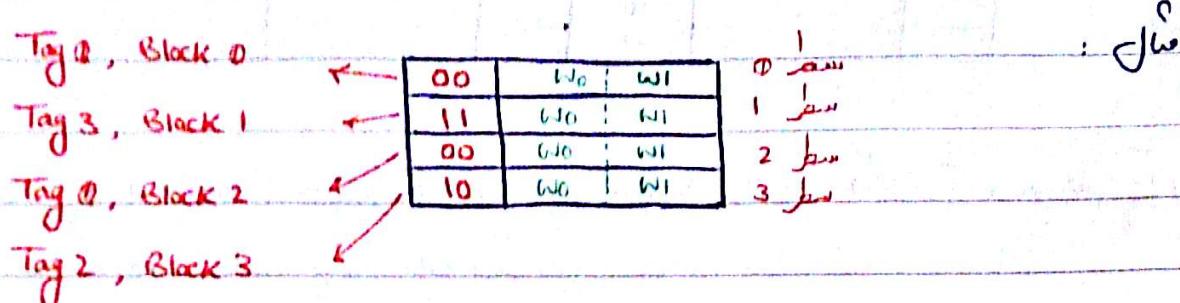
offset : Word-No. (offset)      Tag      Block-No. =  $\log_2$  Block-Size

$\log_2$  Cache-Size

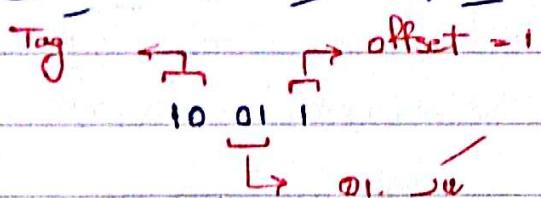
$\log_2$  Main-Size

$$\text{Cache} (= \text{Main}) \times \text{Main-Size} = \text{Cache-Size}$$

$$\log_2 \text{Cache-Size} = \log_2 \# \text{Block} + \log_2 \# \text{Word}$$



لهم اجعلنا في اهل جنات السماوات مقيمين واجعل ملائكة السموات ملائكة مقيمة



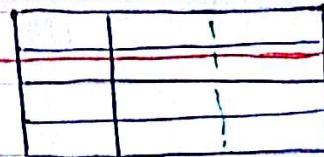
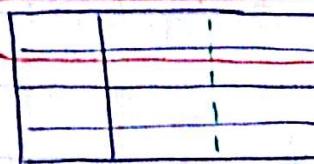
\* عَمَانِي طَهْرَانِي مُدْرَسَتِي بَهْرَانِي بَرَاطَنِي

مثلاً جدول في الذاكرة : Set Associative Mapping (١)

اے سوچیں جس کو اپنے ملک میں نہیں کھانے پڑے۔ Cache میں اسی کا نام رکھ دیں۔

## ( 2-way Set Associative )

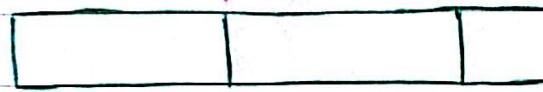
- مفهوم دوگانه Tag 2, Block 0 , Tag 0 , Block 0
- Cache ترکیب از دو قسم Cache ریت و Cache ریپ



Set

\* در این طبقه این ایده را در معرفت سازی (Set) نهاد که در هر مجموعه ۴ جای خالی داشته باشد.

\* مثلاً Cache ۱۶۰ KB حجم ۵۱۲ KB ۴ way ۴ جای خالی داشته باشد که در هر ۴ جای خالی ۱ جای خالی داشته باشد.



$$\log_2 \text{Block-Size} = \log_2 (\text{Set-No.})$$

$$\log_2 \text{Way-Size}$$

$$\log_2 \text{Main-Size}$$

(! این طبقه سطح اول است !!!!) \*\*\* این طبقه سطح اول است !!!

Replacement Policy / خوارق

Cache را کجا می بینیم ؟ پس از ۳۰ ثانیه ؟

اگر Data یا جزو آن را در Cache نداشتم : Random ①

نهایت احتمال داشتم که این داده ها در Cache نداشتم، این روش خوب نیست!

اگر Cache را در Data می بینم : FIFO ②

نهایت احتمال داشتم که این داده ها در Cache نداشتم؛ این روش خوب نیست!

اگر Data را در Cache نداشتم : (Last Recently Used) LRU ③

نهایت احتمال داشتم که این داده ها در Cache نداشتم! و این داده ها از زیر دست داشتم! لذا بحث داشتم که این داده ها در Cache نداشتم!

اگر Data را در Cache نداشتم : (Last Frequently Used) LFU ④

(!) این داده ها در Cache نداشتم. (لذا این داده ها در Cache نداشتم)

با هم خوب نیست!

ازین اینکه FIFO این داده ها را از Cache بخواهد این داده ها در Cache نداشتم!

و همچنان Random این داده ها در Cache نداشتم!

در طبقه Direct این داده ها در Cache نداشتم!

طبقه اوپریک این داده ها در Cache نداشتم!

نهایت احتمال داشتم که این داده ها در Cache نداشتم!

## Write Policy / سیاست نوشتن

اگر مقدار خواندن برابر درست است، اینجا هم خانم از نویس حرف نبینم! صیل  
اگر مقدار خواندن از نویس بیشتر باشد، Cache را نویس نمایم!

اگر مقدار خواندن برابر با مقدار نویس باشد، Cache را در Data را در  
نویس نمایم.

نحوی مقدار داده:

اگر مقدار خواندن برابر با مقدار نویس باشد، Cache را در Data را در  
نویس نمایم: Write-Back ①

اگر مقدار خواندن برابر باشد، Cache را در Data را در  
نویس نمایم!

اگر مقدار خواندن برابر باشد، Cache را در Data را در  
نویس نمایم: Write-Through ②

اگر مقدار خواندن برابر باشد، Cache را در Data را در  
نویس نمایم!

پس از اینکه این Bit را در Cache نویسید، آنرا با زنگ خود  
برای نویسندگان آنرا مطلع کنید.

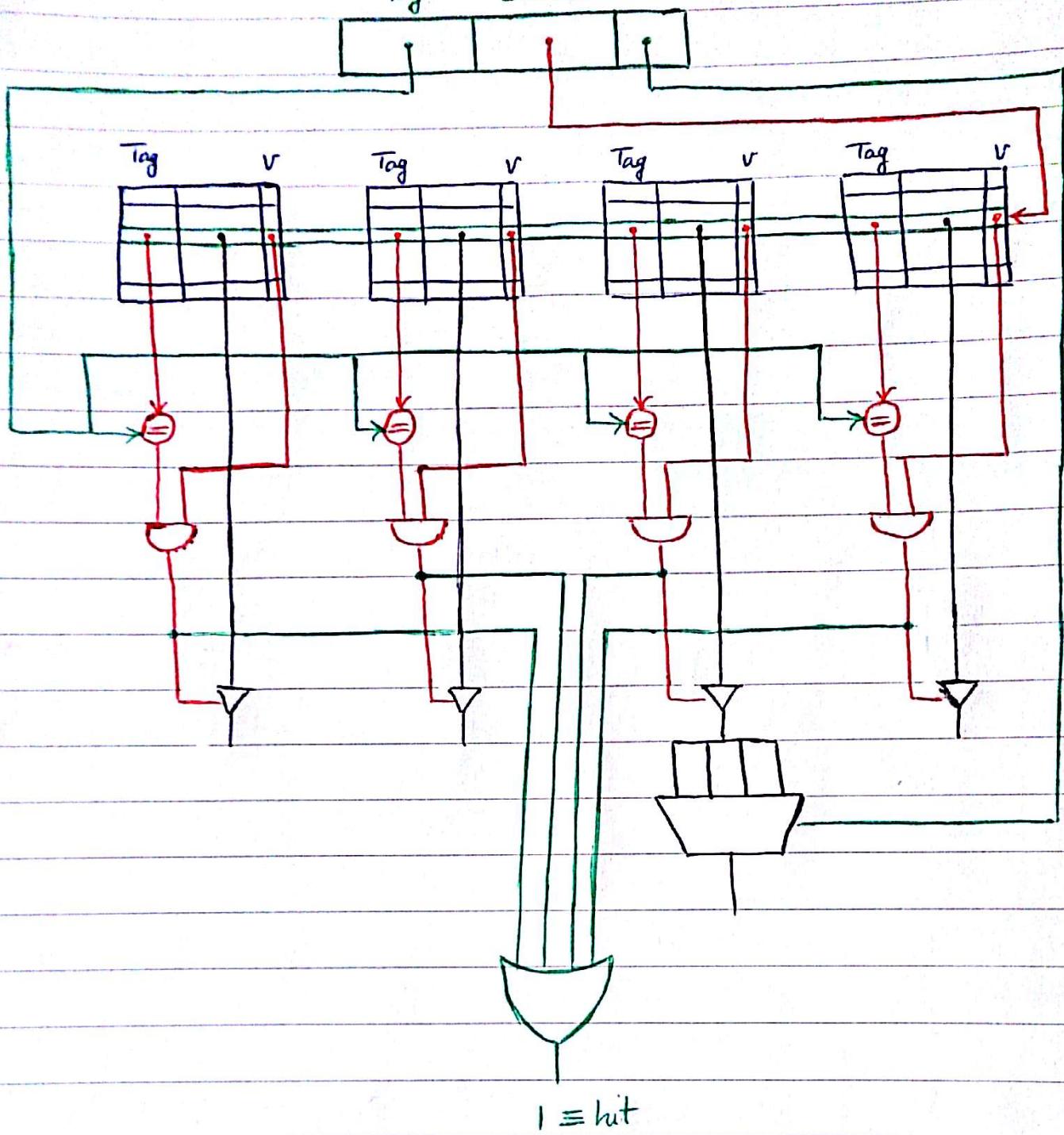
اگر سبب ماموریت این خوبی نهاده شود، Cache را در Data را در  
نویس نمایم!

اگر سبب ماموریت این خوبی نهاده شود، Cache را در Data را در  
نویس نمایم!

DATE / /

SUBJECT: 9r

Tag Set-Nb. word-Nb.



" 4-way Set Associative "