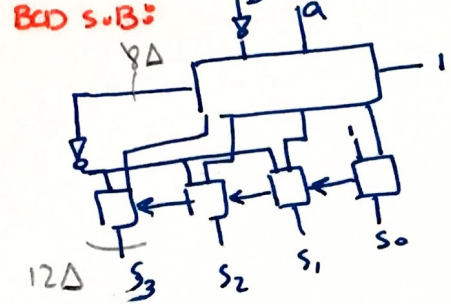
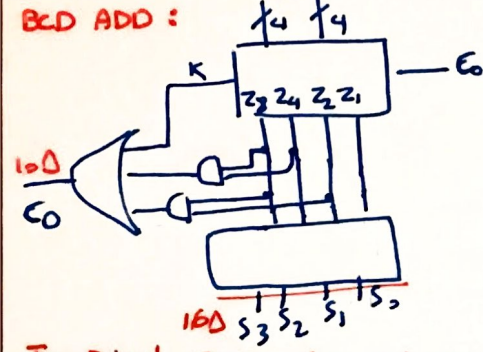


BOD: ADD: A+B

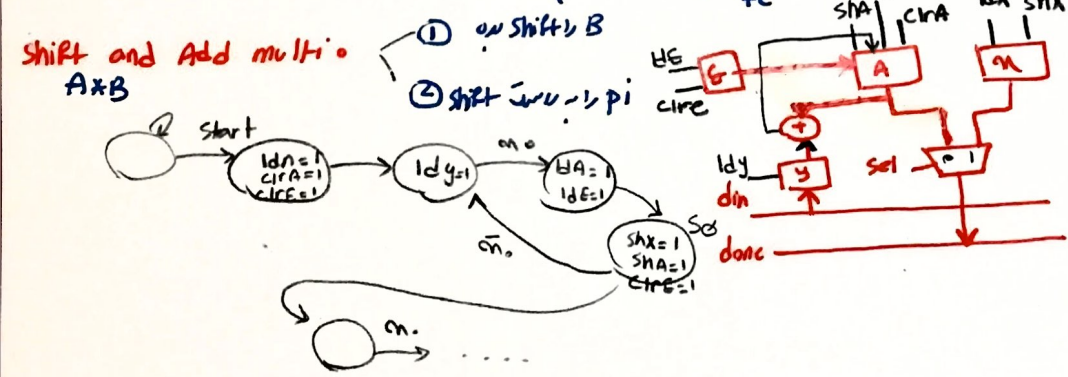
- ① (ex-3) A ② (ex-3) B ③ ADD ④ i_2 $\begin{cases} k=0 & +1101 \\ k=1 & +0011 \end{cases}$
- Sub: A-B
- ② (ex-3) A ② (ex-3) B+1 ③ ADD ⑤ i_2 $\begin{cases} k=0 & +1101 \\ k=1 & +0011 \end{cases}$



T_n -Digit BOD = $(5n+3) 20$

multiplier: $n \times m$

$ts < tc$	$ts \geq tc$	
$t_{end} + (m+n-2)tc$	$t_{end} + mtc + (n-2)tc$	$n > m$
$t_{end} + (m+n-2)tc$	$t_{end} + (m-1)tc + (n-1)tc$	$m > n$



r^n : قدر عدد ϕ : کوچکترین r^{n-1} : بزرگترین r^n : حالت آخر

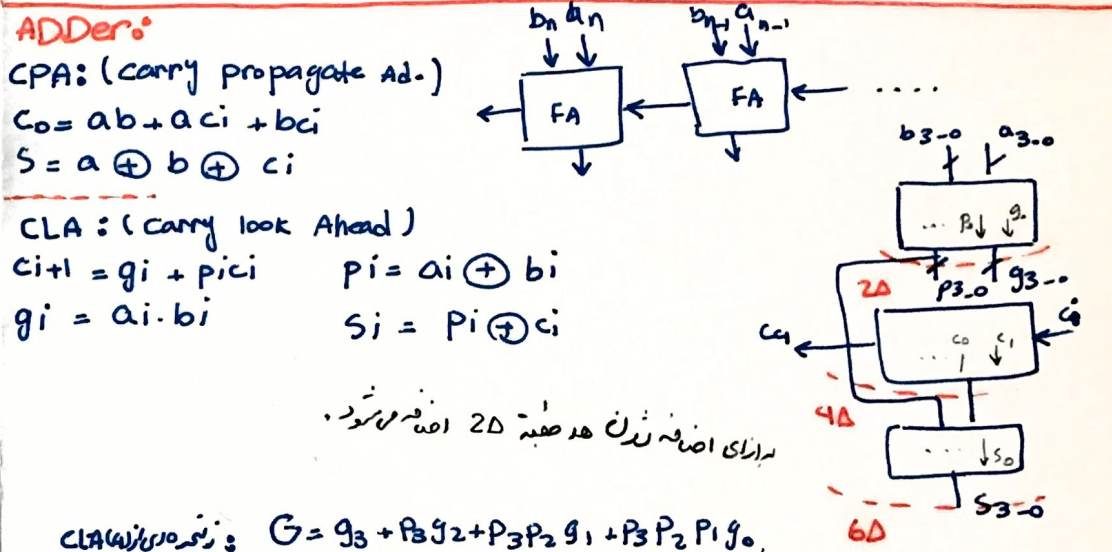
$(r-1)$'s complement: $-N = (r^n - 1) - N$ $-(r^{n-1}-1) < N < (r^{n-1}-1)$

اعداد مثبت به صورت N با هم جمع میشوند و در صورت آید C_0 آن را با حاصل جمع میزنیم.

(r) 's complement: $-N = (r^n) - N$ $-r^{n-1} < N < r^{n-1}$

اعداد مثبت به صورت N با هم جمع میشوند و در صورت آید C_0 آن را با حاصل جمع میزنیم.

عددی که قدر بزرگتر از عدد باشد باید به آن 1 اضافه کرد.



CLA (carry look Ahead):

$G = g_3 + p_3 g_2 + p_3 p_2 g_1 + p_3 p_2 p_1 g_0$

$P = p_3 p_2 p_1 p_0 c_0$

$C_{ig} = G + P c_0 \rightarrow T_{CLA} = (\frac{n}{4} + 2) 20$

CSK (carry skip Adder):

$T_{CSK}(n) = mtc + t_{mux} + (\frac{n}{m} - 2) t_{mux} + (m-1)tc \rightarrow$ worst case delay

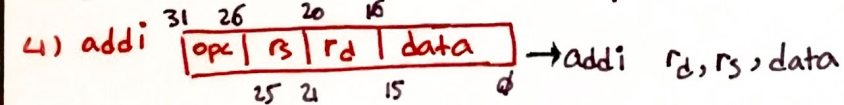
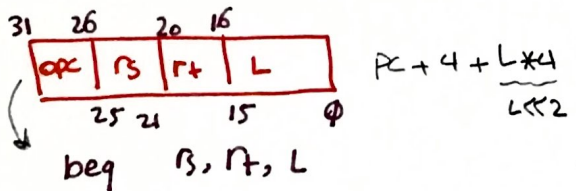
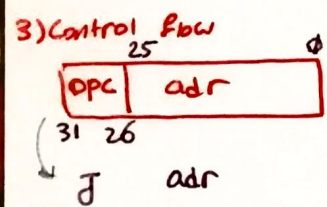
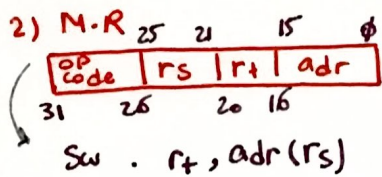
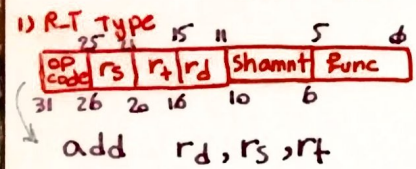
$T_{CSK} = \max((m_i + m_j - 1)tc + (j-i-1)t_{mux}) + t_{mux} + tc$

معماری MIPS:

addu $R_1, R_2, R_3 \rightarrow$ (عدد به دست)

multu $R_1, R_2 \rightarrow$ (ضرب) \rightarrow Hi و Lo (32 bit)

درست کردن MIPS: $2^6 - 1 = 63$ (با R_{10} و R_{11})

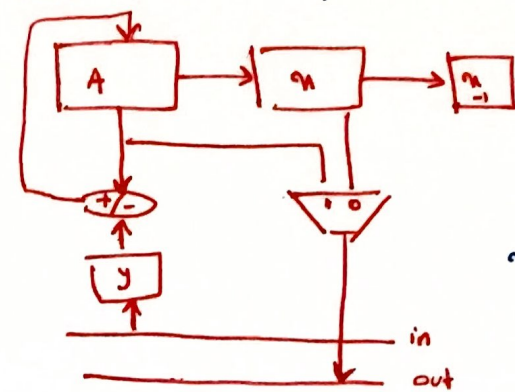


opc	Reg Dst	Reg write	Alu src	mem read	mem write	mem to reg	branch	Alu cp
RT	1	1	0	0	0	0	0	10 \rightarrow RT
Lw	0	1	1	1	0	1	0	00
sw	X	0	1	0	1	X	0	00
beq	X	0	0	0	0	X	0	01
addi	0	1	1	0	0	0	0	00

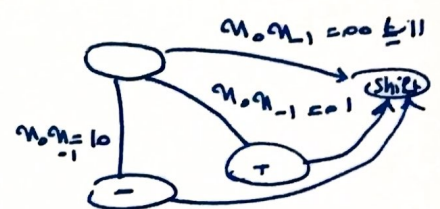
lw/sw

beq

Booth multiplier: $(A \times B)$



1) Both 1B
2) Both 1B
3) Sign extend 1 unit
4) Sign extend 1 unit



Floating point Single $\begin{cases} s=1'b & E=8'b & M=23'b \\ Double & \begin{cases} s=1'b & E=10'b & M=53'b \end{cases} \end{cases}$

$N = (-1)^S \times M \times 2^E$

S | Exponent | M

MIPS دستورات:

Slt (Set less than) : slt R_1, R_2, R_3 (if $R_3 > R_2 \rightarrow R_1 = 1$)

beq (branch equal) : beq R_1, R_2, L

lw (load word) : lw $R_1, \text{adr}(R_2)$

sw (store word) : sw $R_1, \text{adr}(R_2)$

bneq (branch not-equal) : bneq R_1, R_2, L

J (Jump) : J adr

Jr : Jr $R_1 \rightarrow$ آدرس پیش به R_1

push

addi $R_{29}, R_{29}, 4$

sw $R_{31}, \phi(R_{29})$

Jal A \rightarrow بنی لاله

pop

lw $R_{31}, \phi(R_{29})$

addi $R_{29}, R_{29}, -4$

addi $R_i, R_j, \text{عدد}$

add.s F_2, F_4, F_6 Single, Reg 32 بیت است.

add.d F_{10}, F_6, F_{24} Double, Reg 64 بیت است.