

سیستم نمایش اعداد

$$N_v = \sum_{i=-m}^{n-1} a_i r^i, \quad 0 \leq a_i \leq r-1$$

$(\overset{n-1}{\underbrace{a_{n-1} \dots a_0}_{\text{MSD}}}, \overset{-1}{a_{-1}}, \overset{-2}{a_{-2}} \dots \underset{\text{LSD}}{a_{-m}})_r$

سیستم نمایش اعداد نسبی - غیر نسبی

اعداد hex ( $r=16$ ) ←  $\{F, A, B, C, D, E, 0, \dots, 9\}$  و  $a_i \in \{0, 1, \dots, r-1\}$

### تبدیل مبنا

- ① تبدیل از مبنا  $r$  به ۱۰: محاسبه  $N_v$
  - ② تبدیل از مبنا ۱۰ به  $r$ : صیغ ← تقسیم متوالی ← کسری ← ضرب متوالی
  - ③ تبدیل از مبنا  $r_1$  به  $r_2$ : تعداد اعداد ←  $r_2^n$  و  $0 \leq N \leq r_1^n - 1$
- $$(\underbrace{\dots}_{\text{رقم } n_1})_{r_1} \rightarrow (\underbrace{\dots}_{\text{رقم } n_2})_{r_2} \Rightarrow r_1^{n_1} - 1 \leq r_2^{n_2} - 1 \Rightarrow n_2 = \left\lceil \frac{n_1 \log r_1}{\log r_2} \right\rceil$$

### جمع و تفریق

می خواهیم یک مدار واحد برای جمع و تفریق ارائه دهیم

$$A - B = A + (-B)$$

### نمایش اعداد علامت دار (signed Digits)

یک نمایش غیر نسبی ←

$$16 = 1 \times 10 + (-6) \times 1 = 4 \quad -5 = (-1) \times 10 + 5 \times 1 = -5$$

### \* sign-Magnitude \*

منفی  $\rightarrow S = r-1$  و مثبت  $\rightarrow S = 0$

$$N = (S a_{n-2} a_{n-3} \dots a_0)_r \Rightarrow S = 0 \rightarrow \text{مثبت}, S = r-1 \rightarrow \text{منفی}$$

$r=10 \Rightarrow 012 \rightarrow +12$  و  $912 \rightarrow -12$        $r=2 \Rightarrow 0101 \rightarrow 5$  و  $1101 \rightarrow -5$

\* خواسته می مدار محاسباتی واحد را به نتیجه نمی رساند

\* دارای مشکل  $+0$  و  $-0$  می باشد \* بیت علامت جزئی از عدد نیست (بدون محاسبه)

## \*\* جمع و تفریق

① در صورت هم علامت بودن قدرمطلق هاجم می شوند و علامت قبلی گذاشته می شود

② در صورت ناهم علامت بودن قدرمطلق کوچکتر از قدرمطلق بزرگتر کم می شود و علامت قدرمطلق بزرگتر گذاشته می شود

## \* Diminished Radix Comp (r-1)'s Comp

$$N = (0 \ a_{n-2} \ a_{n-3} \ \dots \ a_0)_r \quad -N = r^n - 1 - N \quad -(r^{n-1} - 1) \leq N \leq r^{n-1} - 1$$

\* در تمامی نهائش ها اعداد مثبت با صفر آغاز می شوند. \* بیت علامت جزئی از عدد (با محاسبه)

\* مشکلات: ۱- مشکل +0 و -0. ۲- بار محاسباتی اضافه برای Carry برای

شیوهی تبدیل به قدرمطلق: تمام بیت ها به جز بیت علامت را Reverse کنید.

\* Carry برای مجددا با عدد جمع می شود  $+5 - 3 = +2$

## \* Radix Comp (r's Comp)

$$-N = r^n - N \quad -r^{n-1} \leq N \leq r^{n-1} - 1$$

شیوهی تبدیل به قدرمطلق: تمام صفرها تا اولین یک نگه داشته می شوند و باقی Reverse می شوند.

$$\begin{matrix} 2^3 & 2^2 & 2^1 & 2^0 \\ - & & & \\ 1 & 0 & 1 & 1 \end{matrix}$$

$$1011 = (-2^3) + 2^1 + 2^0 = -8 + 3 = 5$$

\* Carry برای دور ریخته می شود.

\* یک شیوهی دیگر برای تبدیل به ۱۰:

\* مفهوم Overflow: حاصل عملیات بزرگتر از بازه‌ی نهائش اولیه شود  $\leftarrow$  شیوه تشخیص: علامت دو عدد یکسان

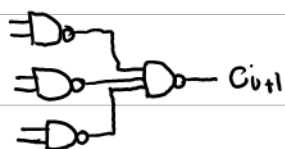
باشد و علامت حاصل باقی ها متفاوت باشد. ۲- Carry ای که به رقم آخر وارد می شود و با Carry ای که خارج می شود یکسان نباشد.

## (مدار جمع کننده)

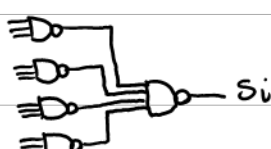
$$C_{i+1} = a_i b_i + a_i C_i + b_i C_i$$

$$S_i = a_i \oplus b_i \oplus C_i = \bar{a}b\bar{c} + \bar{a}\bar{b}c + a\bar{b}\bar{c} + abc$$

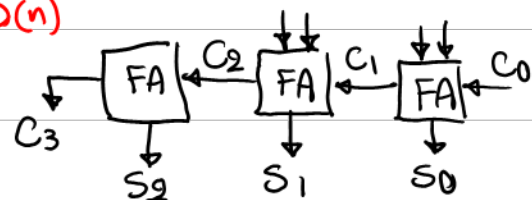
$$\Rightarrow T = n \times 2\Delta = O(n)$$

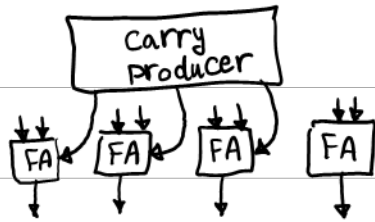


$$T = 2\Delta$$



$$T = 2\Delta \Leftrightarrow \Delta = \text{تاخیر هر گیت}$$

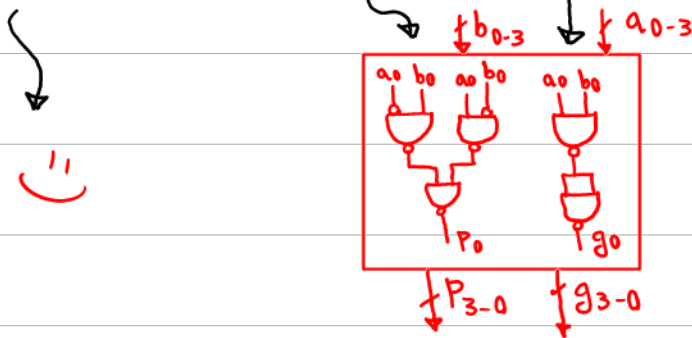




$a_i$	$b_i$	$C_{i+1}$	
0	0	0	→ Kill
0	1	$C_i$	→ propagate
1	0	$C_i$	
1	1	1	→ generate

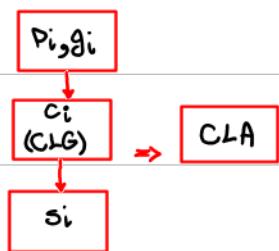
(CLA) Carry Look-Ahead Adder

$$C_{i+1} = g_i + P_i C_i \quad , \quad g_i = a_i \cdot b_i \quad , \quad P_i = a_i \oplus b_i \quad , \quad S_i = a_i \oplus b_i \oplus C_i = P_i \oplus C_i$$



$$T_{CLA}(4) = 6\Delta \quad T_{CPA}(4) = 8\Delta$$

$$T_{CLA}(16) = 12\Delta \quad T_{CPA}(16) = 32\Delta$$



\* مشکل این Adder fan-out و fan-in است (قبل از Cascade)

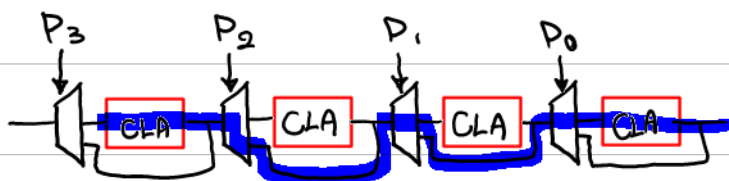
راه حل: Cascading  $\Leftarrow$  CLA(16)  $\leftarrow$  CLA  $\leftarrow$  CLA  $\leftarrow$  CLA

\* در این حالت خود چهار قطعه CLA مجدداً به حالت CLA تبدیل می شوند که  $G_i$  و  $P_i$  تعریف می شوند.

حالات مختلف که یک قطعه Carry تولید کند و بقیه آن Propagate کند.

$$G_4 = g_3 + P_3 g_2 + P_3 P_2 g_1 + P_3 P_2 P_1 g_0 \quad , \quad P_4 = P_3 P_2 P_1 P_0 C_0$$

$$C_4 = G_0 + P_0 C_0 \quad , \quad C_8 = G_1 + P_1 C_4 \Rightarrow T_{CLA} = O(\log_2^n)$$



(CSA) Carry Skip Adder

Critical Pass  $\rightarrow$ 

$$T_{CSK}(n, m) = m \cdot t_c + t_{mux} + \left(\frac{n}{m} - 2\right) t_{mux} + (m-1) t_c + t_s$$

تعداد قطعات هر CLA  
تعداد سیگنال ها

\* در این جمع کسره اگر قرار باشد تمام Adder های یک دسته Carry را انتشار دهند در همان ابتدا Carry توسط mux انتقال می یابد.

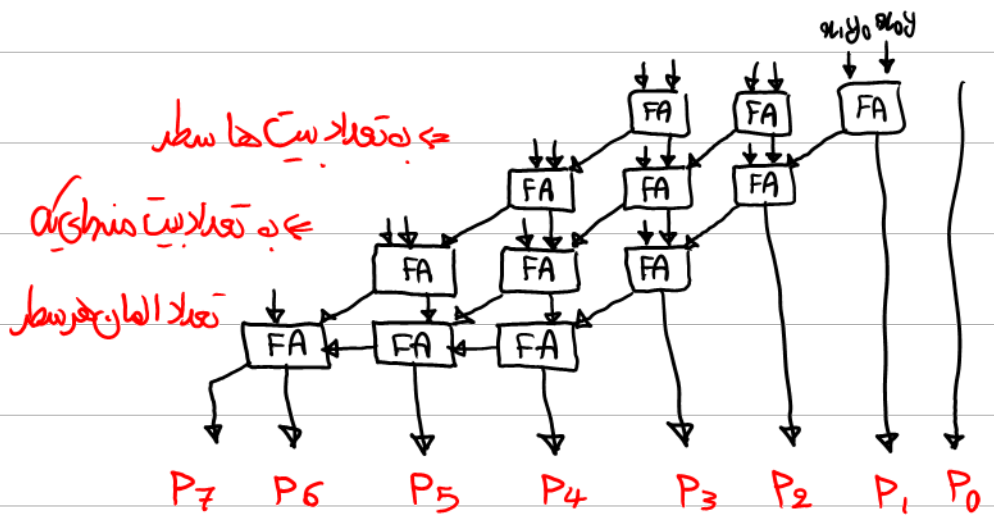
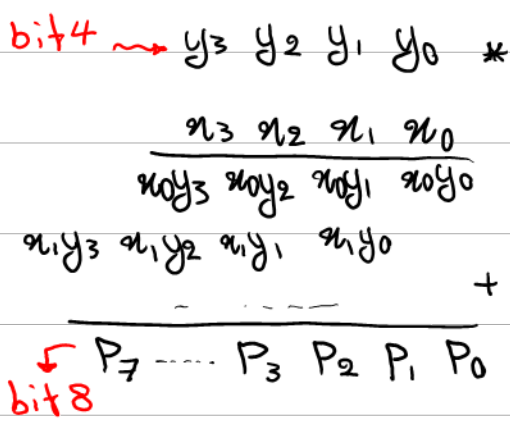
$$\frac{dT(n,m)}{dm} = m_{\text{optimum}} = \sqrt{\frac{t_{\text{max}} \times n}{2tc}}$$

$$\Rightarrow T_{\text{CSK}}(n, m) = O(\sqrt{n})$$

\* می‌توانیم ساینز گروه‌ها (m) را متفاوت در نظر بگیریم که با توجه به مسیر بحرانی باید CLA های اول و آخر را با m کمتر در نظر بگیریم و CLA های میانی را با m بیشتر در نظر بگیریم

۱۴۰۳ / ۷ / ۱۴

### ضرب کسده (Multiplier) - بدون علامت



$2\Delta = \text{تایید FA}$   $\Leftarrow$   $12\Delta = \text{Array Multiplier}$

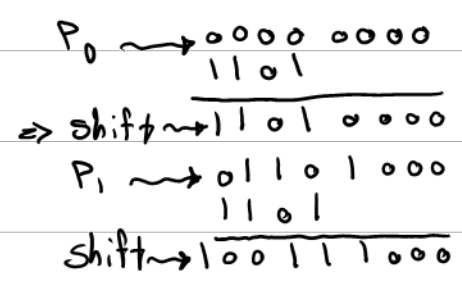
\* Carry ها را به صورت قطری منتقل می‌کنیم. (Carry save Adder) - (به جز ردیف آخر)

### ضرب کسده (Add & shift Multiplier)

$P=0$   
for( $i=0, i<4, i++$ )  
 $P_{i+1} = P_i + x_i 2^i y_i$

الگوریتم  
برجسته

$P=0$   
for( $i=0, i<4, i++$ )  
 $P_i = P_i + x_i y_i$   
 $P_{i+1} = 2^{-1} P_i$

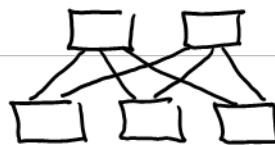


### سیستم دیجیتال سکه‌ها

تمام المان‌های حافظه با یک لبه‌ی CLK کاری‌کنند. \* به دو بخش مسیر داده و کنترل تقسیم می‌شود. مسیر داده، مجموعه‌ای از المان‌های محاسباتی و المان‌های حافظه که توسط BUS به یکدیگر متصل شده‌اند. گذرگاه (BUS): مجموعه‌ای از سیم‌ها که برای انتقال اطلاعات از یک نقطه به نقطه دیگر استفاده می‌شود.

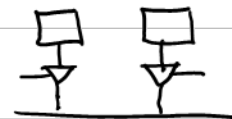


: Common BUS ②



: Dedicated BUS ①

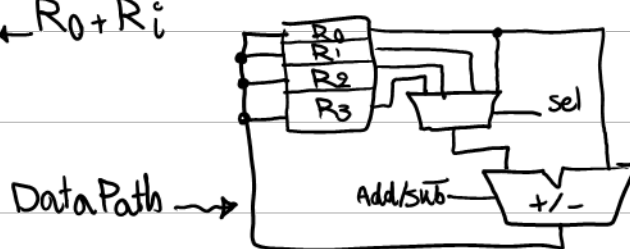
\* در حالت اول امکان به وجود آوردن Z وجود دارد.



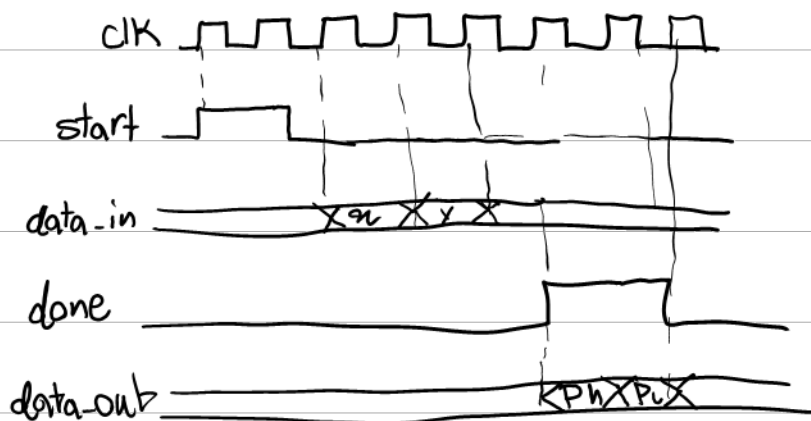
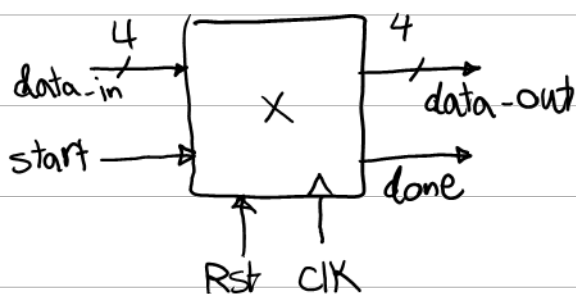
: Common BUS

کلید (کلید)

$$R_j \leftarrow R_0 + R_i$$



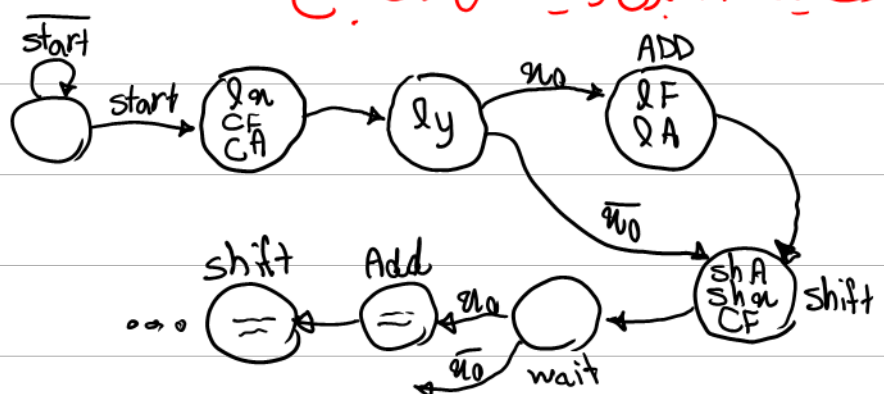
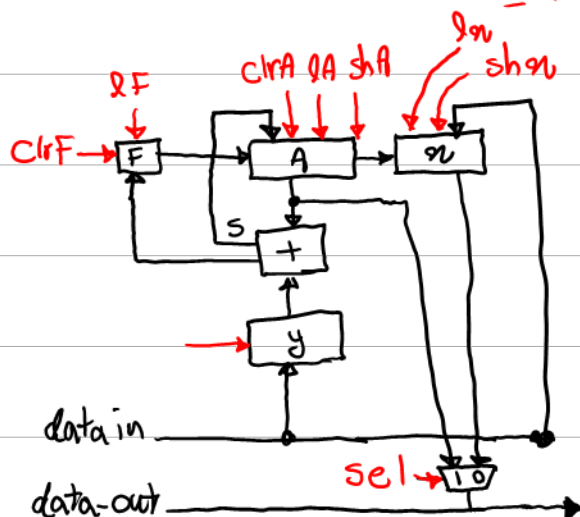
1503/1/19



اجزای مورد نیاز

① جمع کننده 4 بیتی (۲) سه رجیستر 4 بیتی برای مقایسه با قابلیت خود

③ یک FF برای ذخیره Carry جمع



$$30 \times y = 011110 \times y = 32y - 2y = 30y$$

\* از راست به چپ، اولین یک را منفی کنیم سپس اهارا نادیده می گیریم و مجددا اولین صفر را جمع می کنیم

$x_i x_{i-1}$	
NOP	0
ADD	1
SUB	-1 (1)
NOP	0

$$-7 \times -9 = 011001 \downarrow \Rightarrow \begin{array}{r} 11001 \\ \hline 11001 \\ \hline 00111 \\ 00000 \\ 00000 \\ 00000 \\ \hline 11001 \\ 00111 \\ \hline \end{array}$$