«معماری کامپیوتر»

مجید صادقی نژاد

۱۴۰۳/۷/۷

سیستم نمایش اعداد

$(a_{n-1} \dots a_0 . a_{-1} a_{-2} \dots a_{-m})_r$

MSD — LSD

$0 \leqslant a_i \leqslant r-1$

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

سیستم نمایش اعداد سنتی ـ غیرسنتی

اعداد $(r=16)$ hex ⟵ $a_i \in \{0,1,\dots,9,A,B,\dots,F\}$

تبدیل مبنا

① تبدیل ازمبنای r به 10: محاسبه N_v

② تبدیل ازمبنای 10 به r : صحیح ⟵ تقسیم متوالی ، کسری ⟵ ضرب متوالی

③ تبدیل ازمبنای r_1 به r_2: تعداد کل اعداد ⟵ r^n ، $0 \leqslant N \leqslant r^n-1$

$(\underbrace{\quad\quad}_{n_1 \text{ رقم}})_{r_1} \longrightarrow (\underbrace{\quad\quad\quad}_{n \text{ رقم}})_{r_2} \Rightarrow r_1^{n_1}-1 \leqslant r_2^{n}-1 \Rightarrow n = \left\lceil \frac{n_1 \log r_1}{\log r_2} \right\rceil$

جمع و تفریق

می‌خواهیم یک مدار واحد برای جمع و تفریق ارائه دهیم $\quad A-B = A+(-B)$

نمایش اعداد علامت دار ((signed Digit NS))

یک نمایش غیرسنتی ⟵ $\bar{1}5 = (-1)\times 10 + 5\times 1 = -5 \quad\quad 1\bar{6} = 1\times 10 + (-6)\times 1 = 4$

★ sign-Magnitude:

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

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

★ خواسته‌ی مدار محاسباتی واحد را به نتیجه نمی‌رساند $\quad -(r^{n-1}-1) \leqslant N \leqslant (r^{n-1}-1)$

★ دارای مشکل $+0$ و -0 می‌باشد ★ بیت علامت جزئی از عدد نیست (بدون محاسبه)

** جمع وتفریق

① درصورت هم علامت بودن قدرمطلق ها جمع می شوند وعلامت قبلی گذاشته می شود

② درصورت ناهم علامت بودن قدرمطلق کوچکتر از قدرمطلق بزرگتر کم می شود وعلامت قدرمطلق بزرگتر گذاشته می شود.

* ((r-1)'s comp) Diminished Radix Comp

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

☆ در تمامی نمایش 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 نهایی

شیوه ی تبدیل به قدرمطلق: تمام بیت ها به جز بیت علامت را Reverse کنید. (بیت علامت را دور بریزید)

☆ Carry نهایی مجددا با عدد جمع می شود ← $+5-3=+2$

** (r's comp) Radix comp

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

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

☆ Carry نهایی دور ریخته می شود.

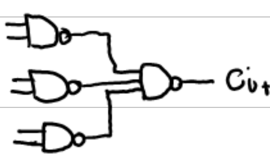
☆ یک شیوه ی دیگر برای تبدیل به 10:

$$\begin{array}{cccc} -2^3 & 2^2 & 2^1 & 2^0 \\ 1 & 0 & 1 & 1 \end{array} = (-2^3)+2^1+2^0=-8+3=5$$

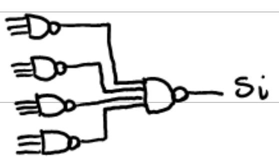
☆ مفهوم overflow: حاصل عملیات بزرگتر از بازه ی نمایش اولیه شود ← شیوه ی تشخیص، علامت دو عدد یکسان باشد و علامت حاصل با آن ها متفاوت باشد ۲- Carry ای که به رقم آخر واردی شود و با Carry ای که خارج می شود یکسان نباشد

مدار جمع کننده

$C_{i+1}=a_ib_i+a_ic_i+b_ic_i$ $\qquad S_i=a_i\oplus b_i\oplus C_i=\bar{a}\bar{b}c+\bar{a}b\bar{c}+a\bar{b}\bar{c}+abc$

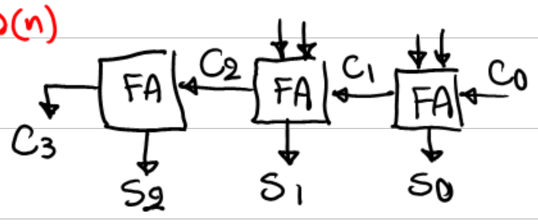


$T=2\Delta$



$T=2\Delta \Leftarrow \Delta=$ تاخیر هرگیت

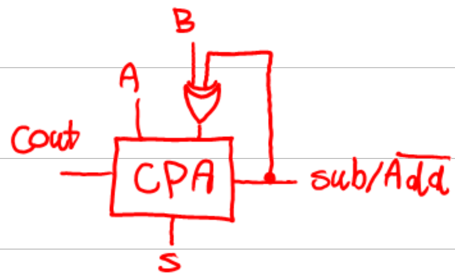
$\Rightarrow T=n\times 2\Delta=O(n)$



مدار جمع و تفریق کننده با 2's Comp)

$$A - B = A + (-B) = A + (2's\,Comp(B)) = A + (1's\,Comp(B) + 1) = A + NOT(B) + 1$$

Half adder مبحث ۲



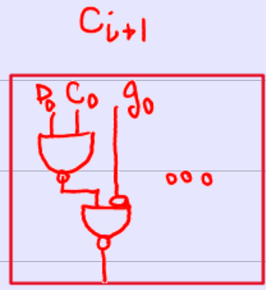
CLA)

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

$C_1 = g_0 + P_0 C_0$

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

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



$T(g_i) = T(P_i) = 2\Delta \Rightarrow T(C_{i+1}) = 4\Delta \Rightarrow T(S) = 6\Delta$

محاسبه تأخیر CLA به فرم CLA ها!)

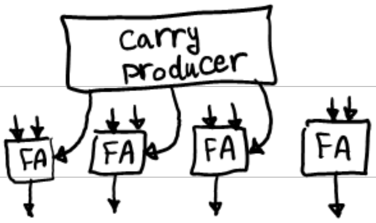
۱) محاسبه p_i و g_i ها ← 2Δ ۲) محاسبه P_i و G_i ها ← 2Δ

۳) محاسبه‌ی C_4 و C_8 و C_{12} و ... ← 2Δ ۴) محاسبه‌ی باقی C_i های (درونی) ← 2Δ

۵) محاسبه‌ی S_i ← 2Δ ⇐ مجموعاً 10Δ

۱۴۰۳/۷/۹

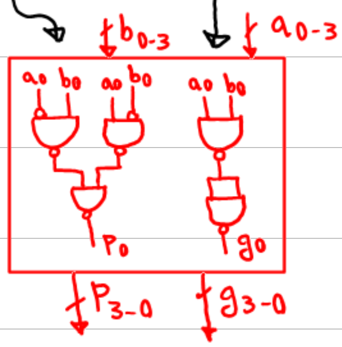
(CLA) Carry Look-Ahead Adder



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

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

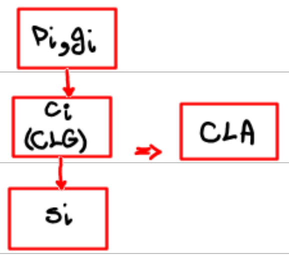
در صفحه‌ی قبل ببینید



در صفحه‌ی قبل ببینید

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

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



★مشکل این Adder مبحث fan-in، fan-out در جمع کننده‌های بزرگ است (قبل از Cascade)

راه حل: cascading ⇐ CLA(16) CLA ← CLA ← CLA ← CLA

★در این حالت خود چهار قطعه CLA مجددا به حالت CLA تبدیل می‌شوند که G_i و P_i تعریف می‌شوند.

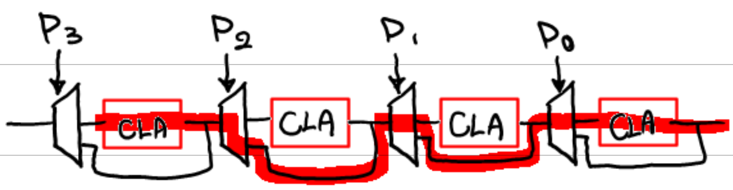
★ در این حالت تأخیر CLA(16) به 10Δ می‌رسد ← توضیحات بیشتر در صفحه‌ی قبل

حالات مختلفی که یک قطعه carry تولید کند و بقیه آن را Propagate کنند.

$G_0 = g_3 + P_3 g_2 + P_3 P_2 g_1 + P_3 P_2 P_1 g_0$ ، $P_0 = P_3 P_2 P_1 P_0$

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

(CSA) Carry skip Adder



Critical Pass ⤳

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

تعداد بیت‌ها

تعداد قطعات هر CLA

★در این جمع کننده اگر قرار باشد تمام Adder های یک دسته carry را انتشار دهند در همان ابتدا carry توسط mux انتقال می‌یابد.

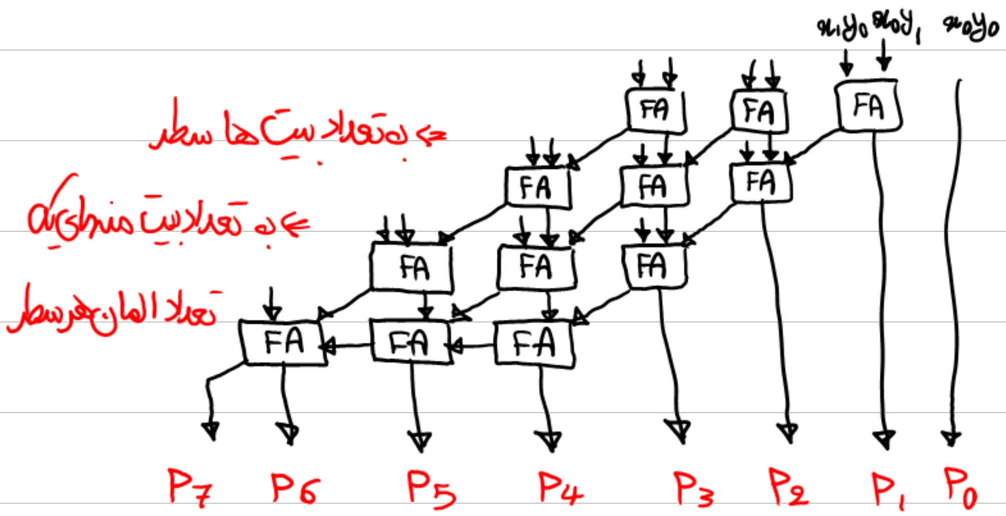
$$\frac{dT(n,m)}{dm} = m_{optimum} = \sqrt{\frac{t_{max}}{2t_c} \times n} \quad \Rightarrow T_{csk}(n,m) = O(\sqrt{n})$$

* می توانیم سایز گروه ها (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 کننده (Multiplier) - بدون علامت)

$$\begin{array}{rrrrrrrr}
\text{bit4} \rightsquigarrow & & & & y_3 & y_2 & y_1 & y_0 \quad * \\
 & & & & x_3 & x_2 & x_1 & x_0 \\
\hline
 & & & & x_0y_3 & x_0y_2 & x_0y_1 & x_0y_0 \\
 & & & x_1y_3 & x_1y_2 & x_1y_1 & x_1y_0 & \\
 & & \cdots & & & & & + \\
\hline
\text{bit8} & P_7 & \cdots & & P_3 & P_2 & P_1 & P_0
\end{array}$$



⇐ تاخیر FA = 2Δ ⇐ تاخیر Array Multiplier = 12Δ

* carry ها را بصورت قطری منتقل می کنیم. (carry save Adder) - (به جز ردیف آخر)

ضرب کننده (Add & shif Multiplier

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

الگوریتم بهتر ⟶

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

$$\begin{array}{rl}
P_0 \rightsquigarrow & 0000\ 0000 \\
 & 1101 \\
\hline
\Rightarrow \text{shift} \rightsquigarrow & 1101\ 0000 \\
P_1 \rightsquigarrow & 0110\ 1000 \\
 & 1101 \\
\hline
\text{shift} \rightsquigarrow & 1001\ 1100\,0
\end{array}$$

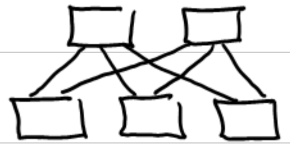
سیستم دیجیتال سنکرون)

تمام المان های حافظه با یک لبه ی CLK کار می کنند. * به دو بخش مسیر داده و کنترلر تقسیم می شود

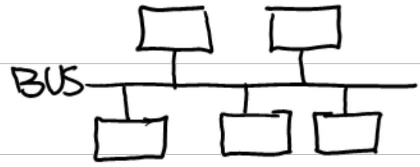
مسیر داده: مجموعه ای از المان های محاسباتی و المان های حافظه که توسط BUS به یکدیگر متصل شده اند.

گذرگاه (BUS): مجموعه ای از سیم ها که برای انتقال اطلاعات از یک نقطه به نقطه دیگر استفاده می شود

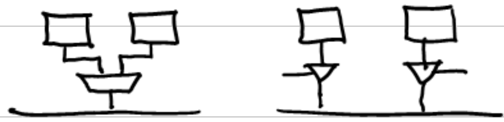
① Dedicated BUS:



② Common BUS:



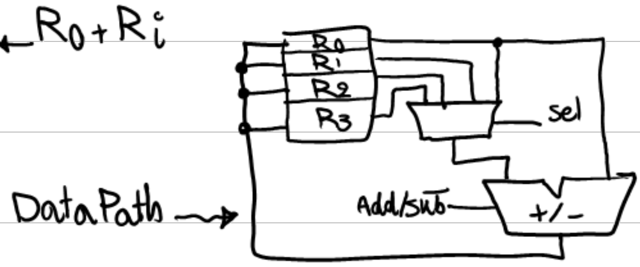
Common BUS:



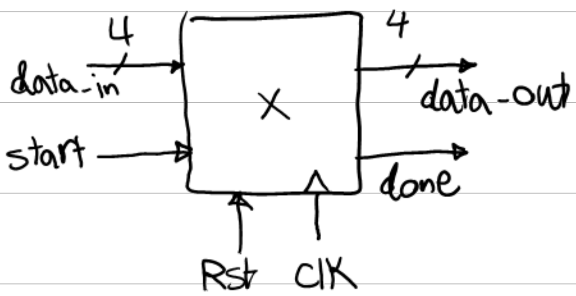
* در حالت اول امکان ① به وجود آوردن Z وجود دارد.

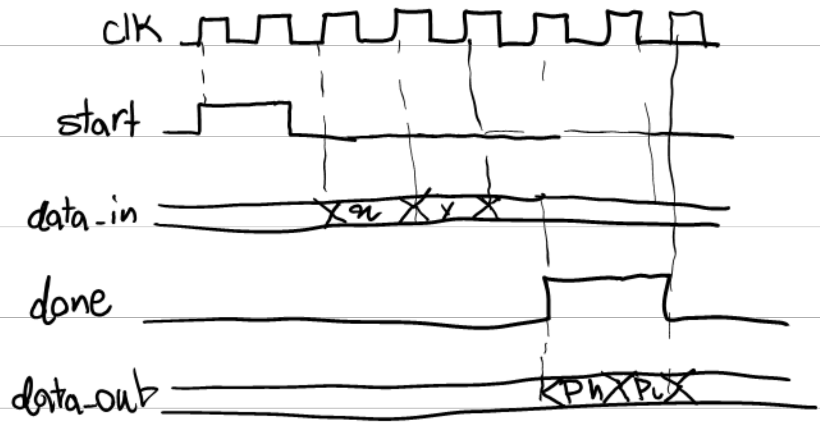
یک مثال:

$R_j \leftarrow R_0 + R_i$



۱۴۰۳/۷/۱۶

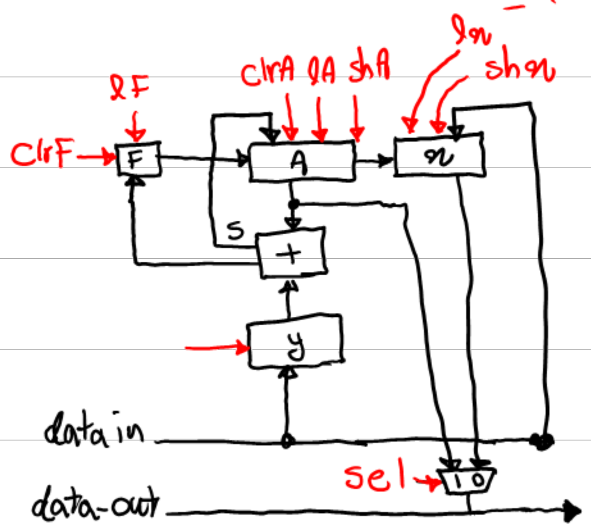


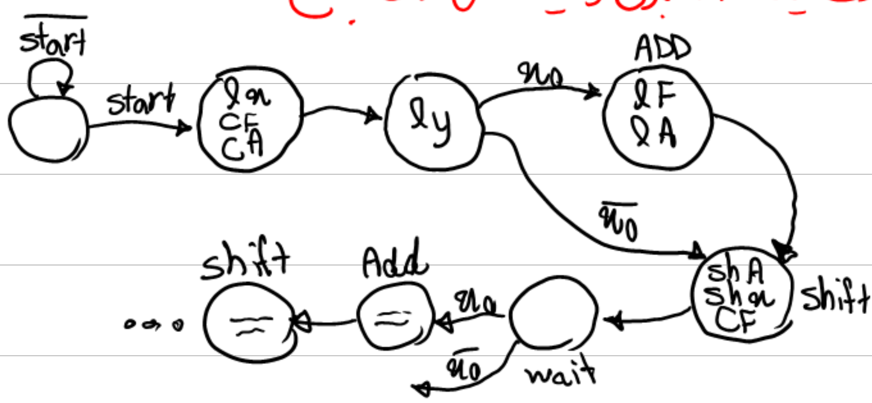


اجزای مورد نیاز:

① جمع کننده ۴ بیتی ② سه رجیستر ۴ بیتی برای ورودی با قابلیت لود

③ یک FF برای ذخیره Carry جمع





Booth Multiplier

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

* از راست به چپ، اولین یک را منفی می کنیم سپس ۱ ها را نادیده می گیریم و مجددا اولین صفر را جمع می کنیم

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

$-7 \times -9 = \begin{array}{c} 0\,11001 \\ 10111\,0 \end{array} \Rightarrow$

$$
\begin{array}{r}
11001 \\
\bar{1}100\bar{1} \\
\hline
00111 \\
00000 \\
00000 \\
-1\;\;11001 \\
00111 \\
\hline
\end{array}
$$