



# معماری و سازمان کامپیوتر

دانشگاه صنعتی اصفهان

دانشکده مهندسی برق و کامپیوتر

امیر خورسندی

بهار ۱۴۰۲

# محاسبات در کامپیوتر

## مقدمه

- می توان گفت مهم ترین وظیفه کامپیوتر انجام محاسبات و پردازش داده ها است.
- اگرچه شاید لزوماً مهم ترین بخش یک کامپیوتر واحد پردازش و محاسبات نباشد.

# انواع محاسبات

- برخی از محاسبات ساده و سریع هستند:

- And, Or, Xor

- برخی دیگر نیاز به زمان بیشتری به منظور حصول نتیجه دارند:

- جمع و تفریق

- ضرب و تقسیم

- محاسبات اعشاری

- توابع محاسباتی و مثلثاتی

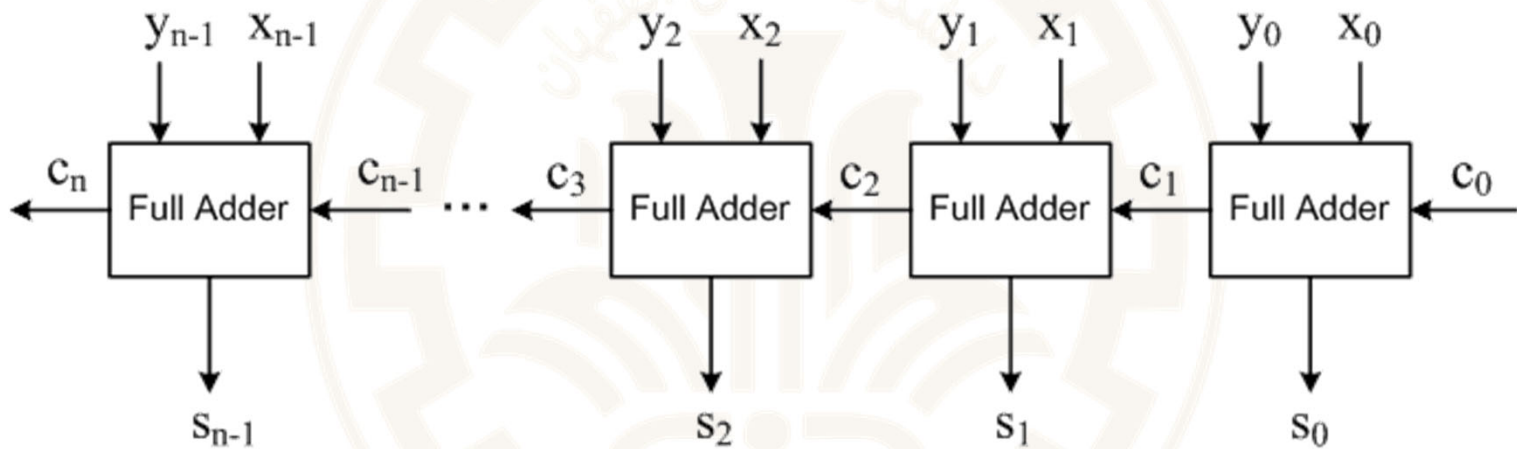
# طراحی سیستم

- در سیستم همزمان، دوره تناوب پالس ساعت متناسب با کندترین عملیات تعیین می شود.
- همیشه سایر بخش ها معطل کندترین واحد خواهد بود.
- سیستم غیرهمزمان پیچیده و پرهزینه خواهد بود.
- در طراحی پردازنده با سرعت و فرکانس کاری بالا سادگی و کارایی بیشتر مدنظر است.

# نتیجه گیری

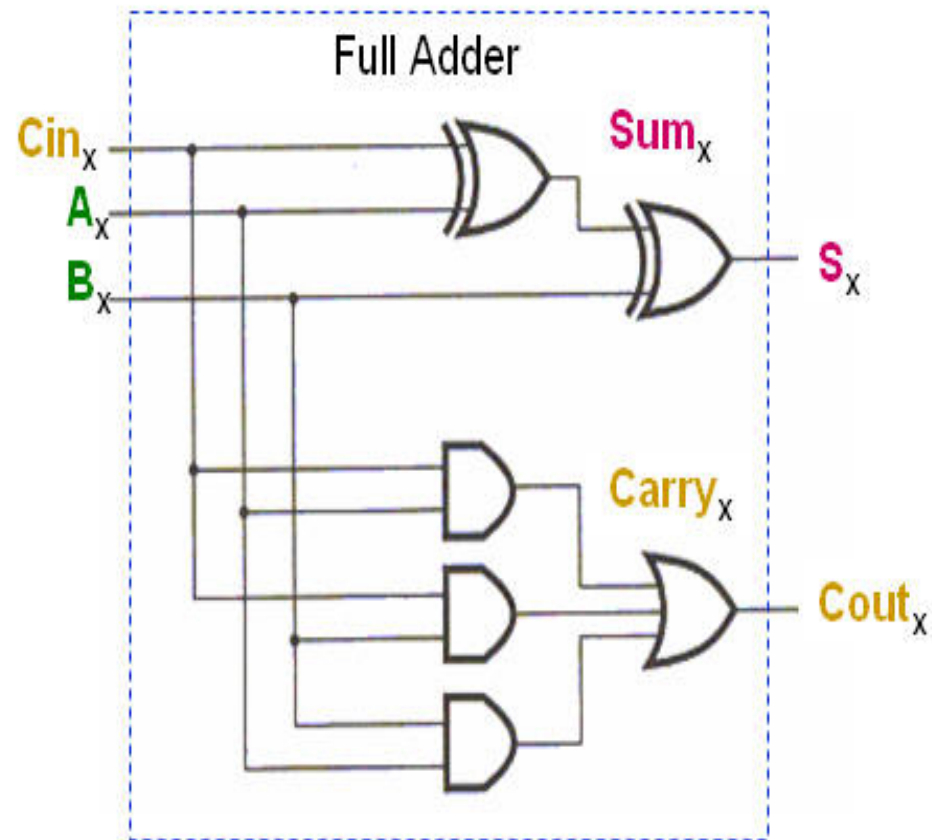
- باید راهکاری مورد استفاده قرار بگیرد که:
- تاخیر عملکرد همه بخش ها و واحدها تا حد ممکن به یکدیگر نزدیک شود.
- فرکانس کاری سیستم افزایش یابد.
- راندمان سیستم افزایش یابد.

# مدار جمع کننده



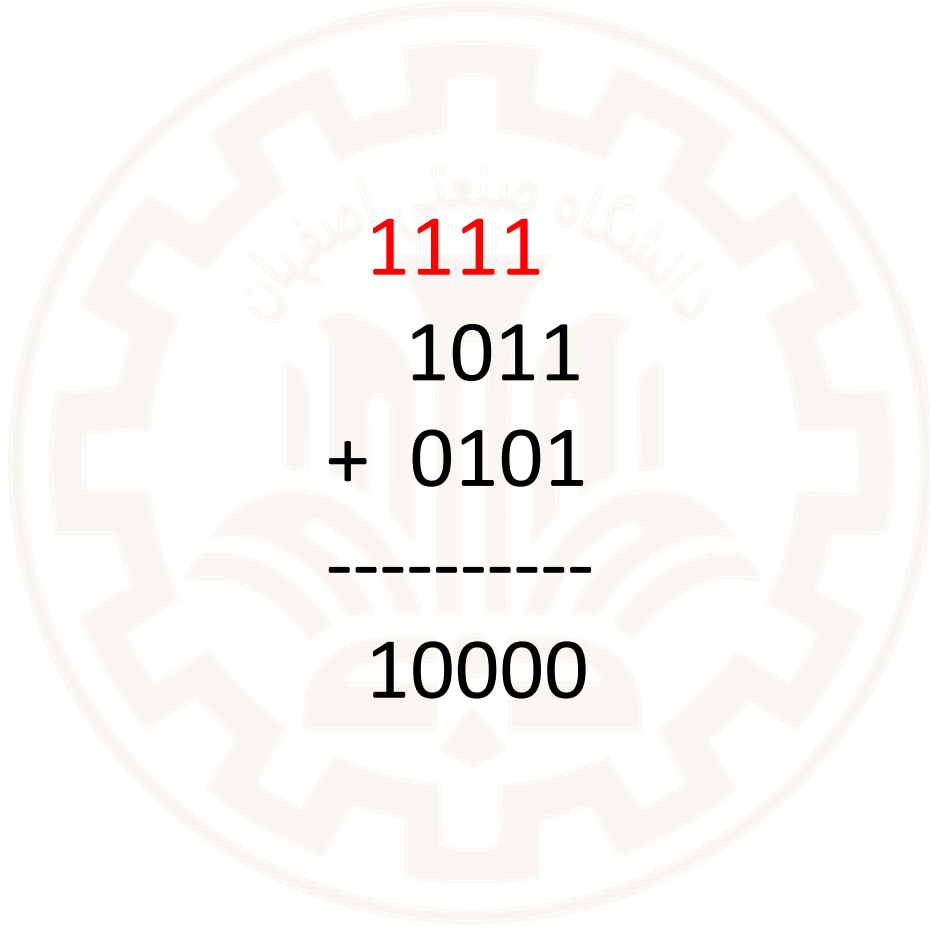
Ripple Carry Adder

# مدار جمع کننده (ادامه)

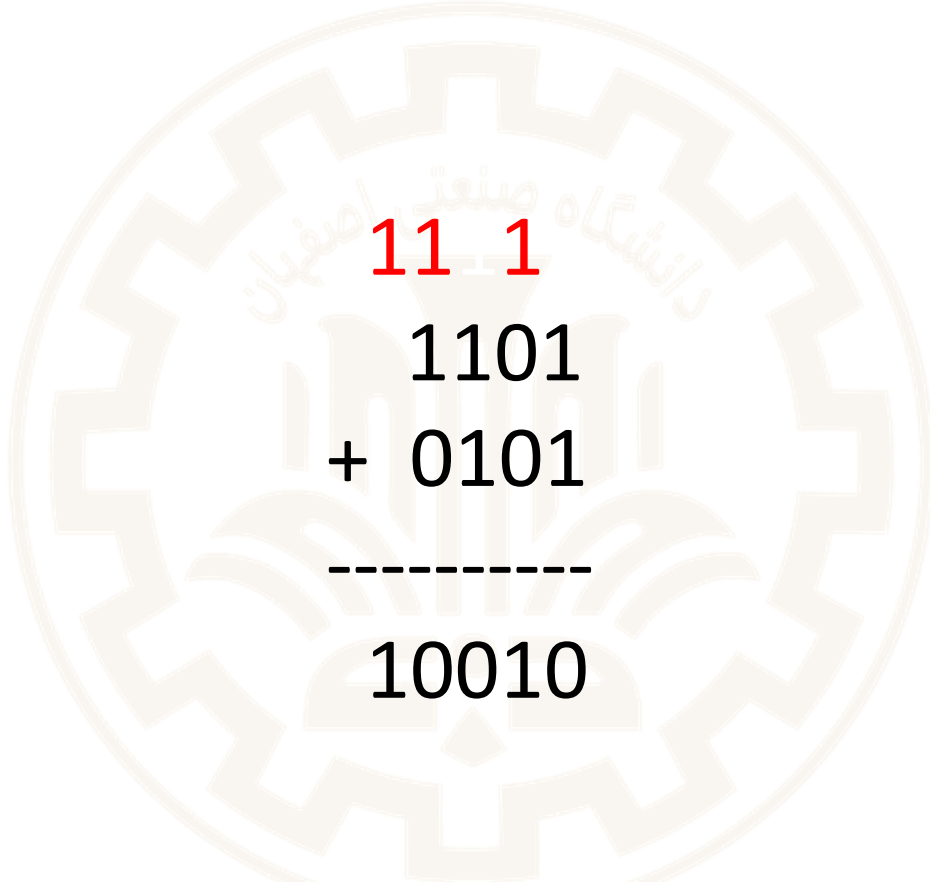




# مثال بدترین تاخیر


$$\begin{array}{r} 1111 \\ 1011 \\ + 0101 \\ \hline 10000 \end{array}$$

# مثال


$$\begin{array}{r} 11\ 1 \\ 1101 \\ +\ 0101 \\ \hline 10010 \end{array}$$

# مدار جمع کننده با پیش بینی بیت نقلی

• دو سیگنال  $g_i$  و  $p_i$  به صورت زیر تعریف می شوند:

◦  $x_i + y_i = 2$  اگر و تنها اگر  $g_i = 1$

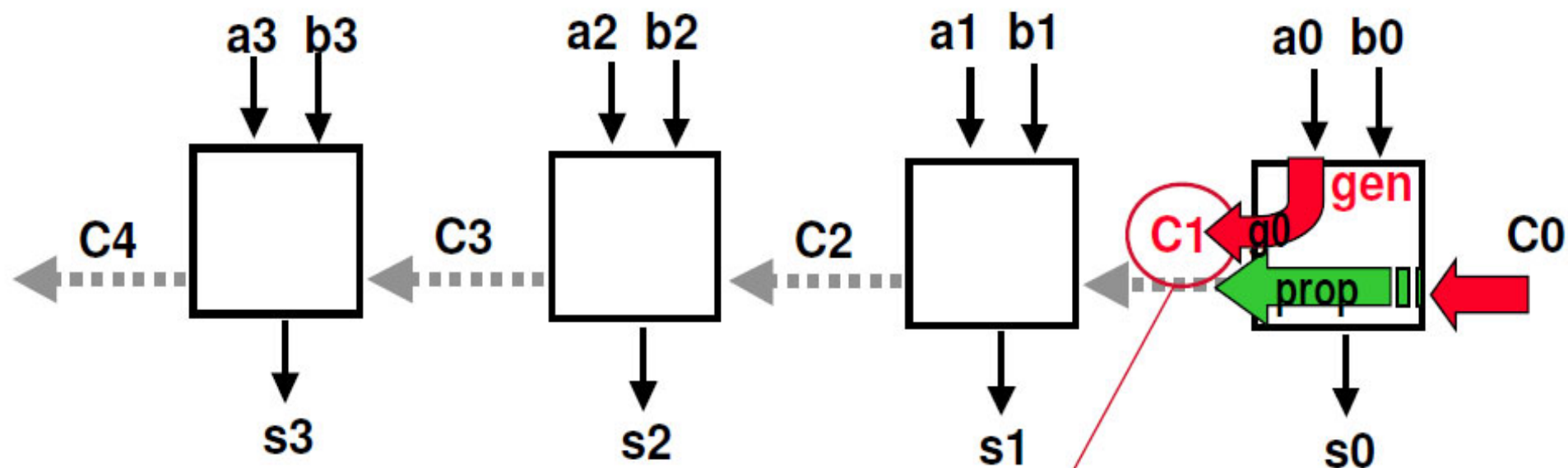
◦  $x_i + y_i = 1$  اگر و تنها اگر  $p_i = 1$

# مدار جمع کننده با پیش بینی بیت نقلی (ادامه)

• بر این اساس بیت نقلی در هر مرحله به صورت زیر محاسبه می شود:

$$\begin{aligned}c_i &= g_{i-1} + c_{i-1}p_{i-1} \\&= g_{i-1} + (g_{i-2} + c_{i-2}p_{i-2})p_{i-1} \\&= g_{i-1} + (g_{i-2} + (g_{i-3} + c_{i-3}p_{i-3})p_{i-2})p_{i-1} \\&= \dots\end{aligned}$$

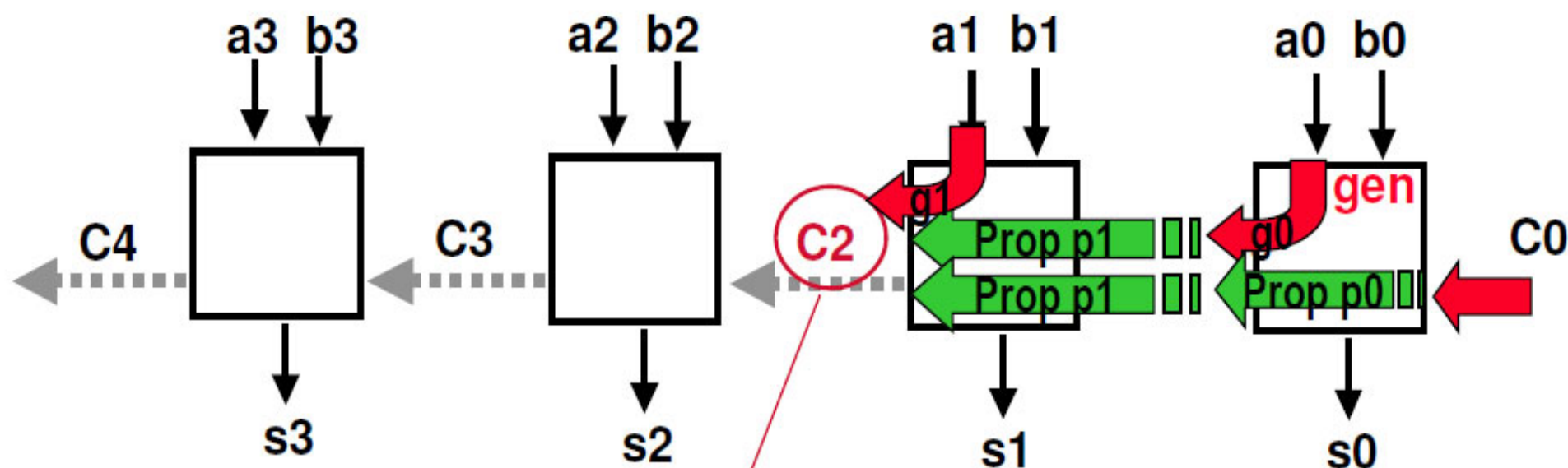
# مدار جمع کننده با پیش بینی بیت نقلی (ادامه)



$$C1 = g0 + poC0$$

[پردازنده های محاسباتی - پروفسور شادرخ سماوی]

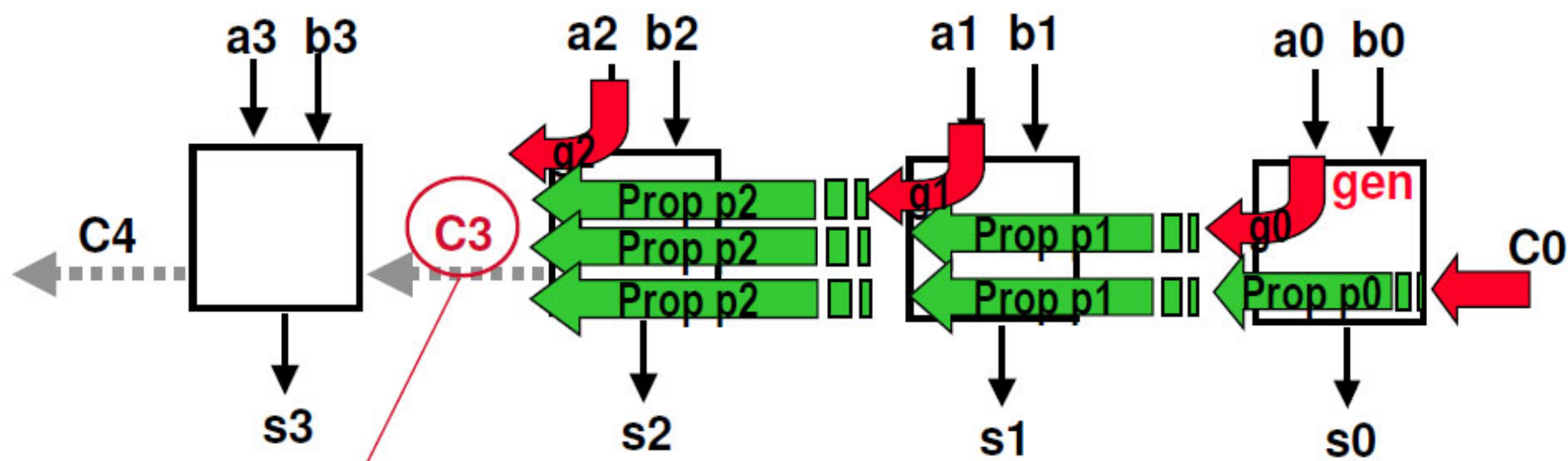
# مدار جمع کننده با پیش بینی بیت نقلی (ادامه)



$$C2 = g1 + p1g0 + p1p0C0$$

[پردازنده های محاسباتی - پروفسور شادرخ سماوی]

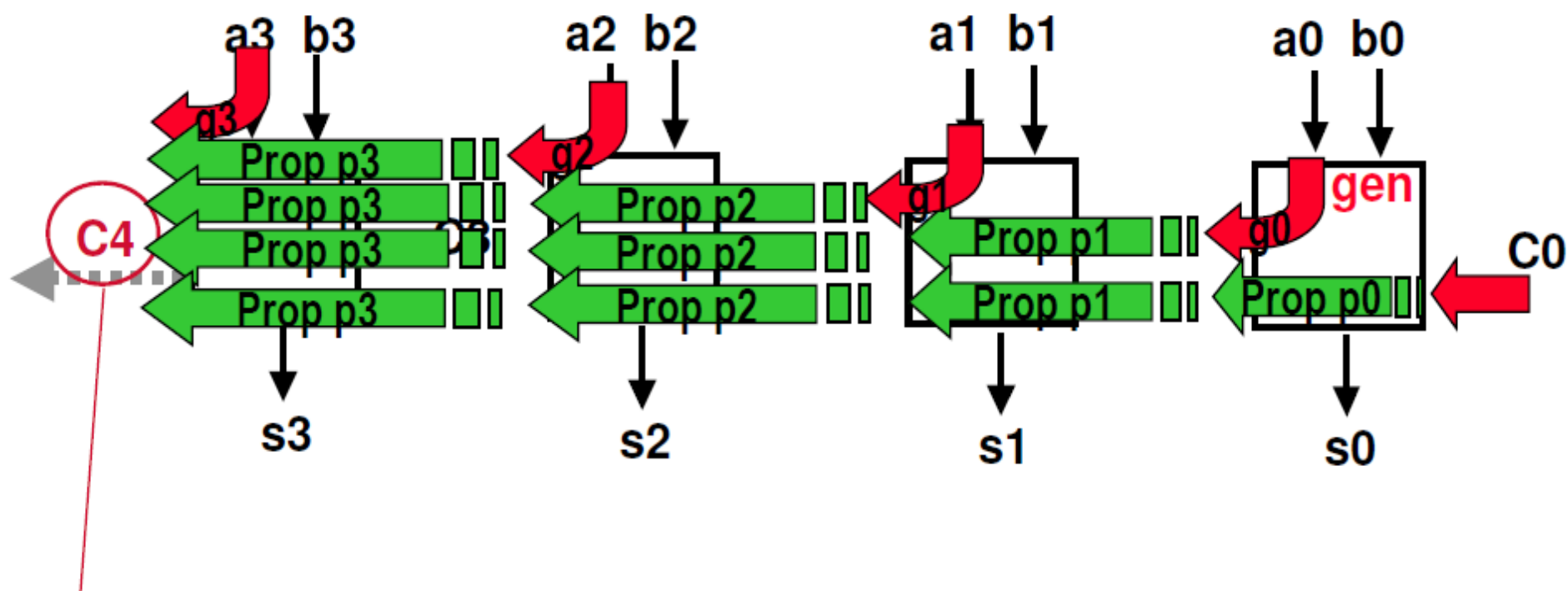
# مدار جمع کننده با پیش بینی بیت نقلی (ادامه)



$$C3 = g2 + p2g1 + p2p1g0 + p2p1p0C0$$

[پردازنده های محاسباتی - پروفسور شادرخ سماوی]

# مدار جمع کننده با پیش بینی بیت نقلی (ادامه)

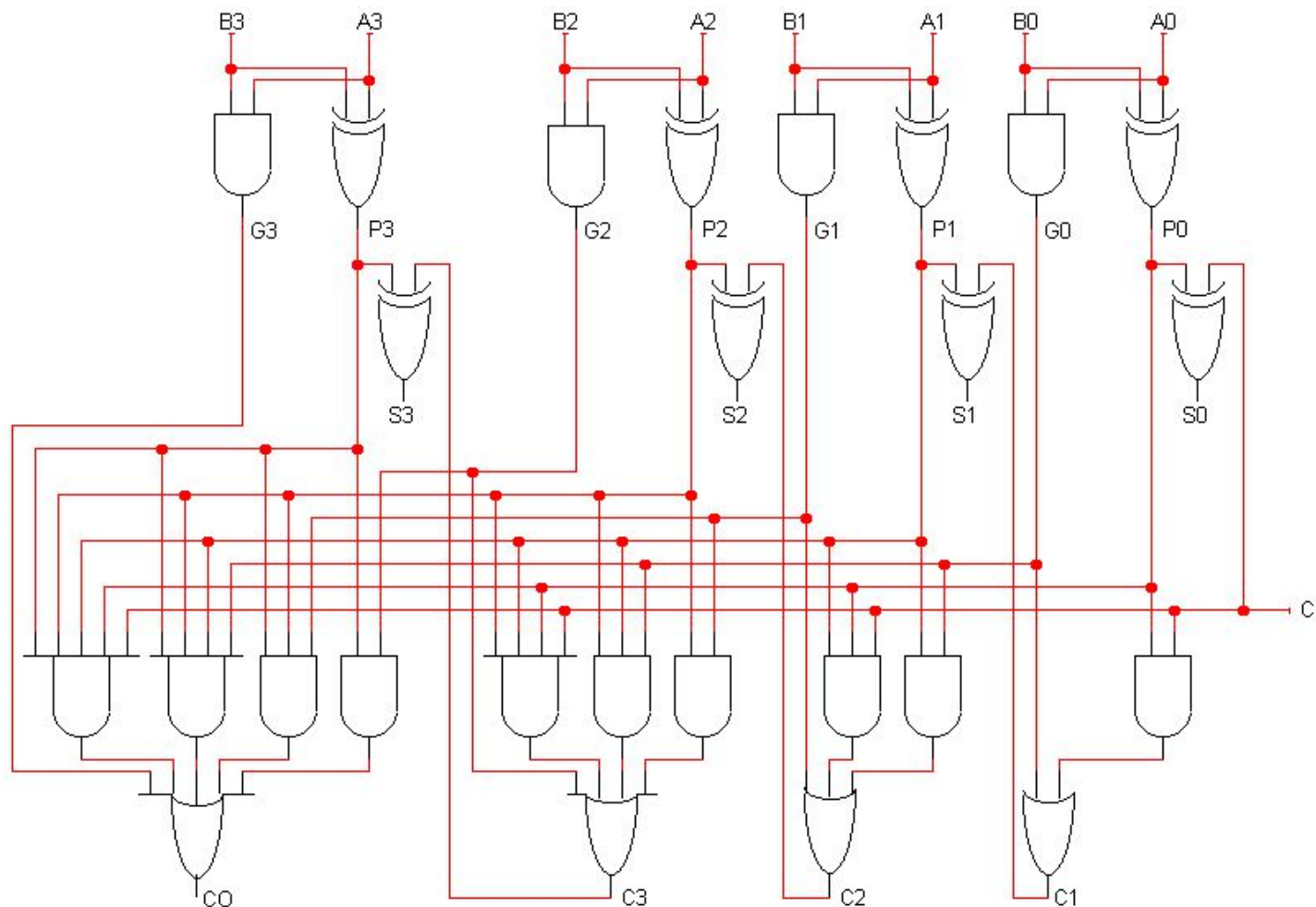


$$C4 = g3 + p3g2 + p3p2g1 + p3p2p1g0 + p3p2p1p0C0$$

[پردازنده های محاسباتی - پروفسور شادرخ سماوی]

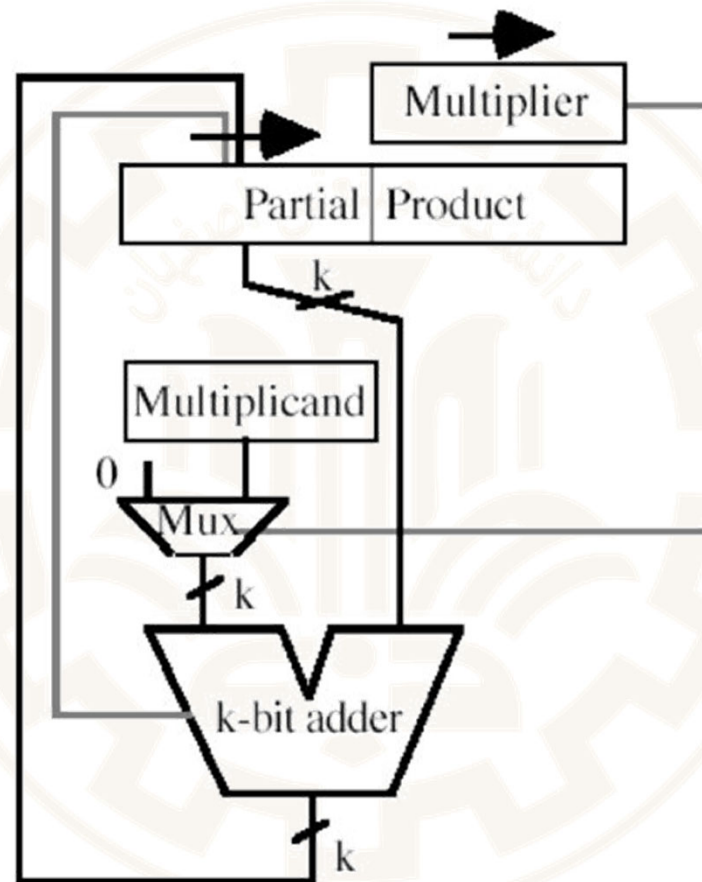


# مدار جمع کننده با پیش بینی بیت نقلی (ادامه)



Carry Lookahead Adder

# مدار ضرب کننده



# ضرب کننده Booth

- در حالت عادی به تعداد بیت های یک موجود در مضروب فیه باید عملیات جمع انجام شود.
- ایده کلی جهت افزایش سرعت:  
با کد کردن مضروب فیه به صورت خاص تعداد عملیات جمع را کاهش دهیم.

# مثال

$$\begin{array}{r} 10010111 \\ \times 10111110 \\ \hline \end{array}$$

نتیجه برابر است با:

$$A \times (-2^7 + 2^5 + 2^4 + 2^3 + 2^2 + 2)$$

# قاعده ریاضی

$$2^j + 2^{j-1} + \dots + 2^{i+1} + 2^i = 2^{j+1} - 2^i$$

# مثال به روش Booth

$$\begin{array}{r} 10010111 \\ \times 10111110 \\ \hline \end{array}$$

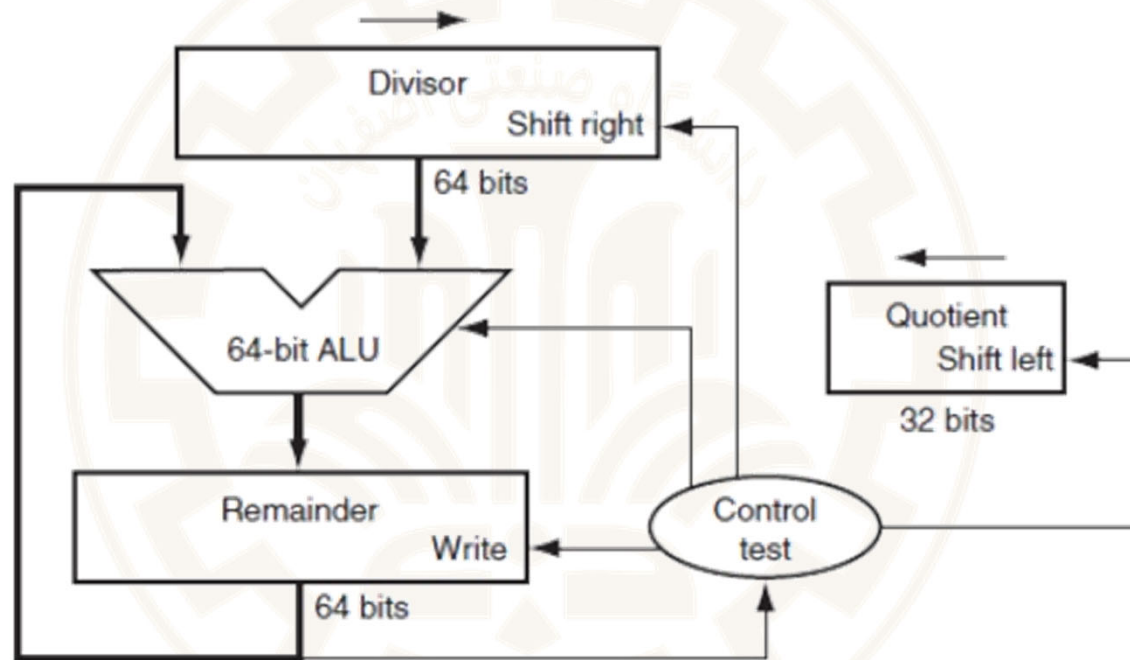
مضروب فیه به صورت زیر کد می شود:

$$\begin{array}{r} 10010111 \\ \times \bar{1}10000\bar{1}0 \\ \hline \end{array}$$

نتیجه برابر است با:

$$A \times (-2^7 + 2^6 - 2)$$

# مدار تقسیم کننده



# محاسبات اعشاری

• ممیز ثابت

• ممیز شناور





# پردازش اعداد اعشاری ممیز ثابت

$x = 000000000.00001001$

$y = 10010000.000000000$

- $y^2$  را به دلیل Overflow نمی توان نمایش داد.
- $x^2$  را به دلیل Underflow نمی توان نمایش داد.

# جمع و تفریق اعداد اعشاری ممیز شناور

$$x = s.2^e.(1+m)$$

$$y = s'.2^{e'}.(1+m')$$

$$x + y = ?$$

- تراز کردن توان ها
- جمع/تفریق کردن مانتیس ها
- نرمال کردن مانتیس ها

$$x + y = [s.(1+m) + s'(1+m')/2^{(e-e')}] . 2^e = s''.(1+m'').2^{e''}$$

# ضرب و تقسیم اعداد اعشاری ممیز شناور

$$x = s.2^e.(1+m)$$

$$y = s'.2^{e'}.(1+m')$$

$$x \times y = ?$$

$$x \times y = (s \times s').2^{(e+e')}.[(1+m) \times (1+m')]$$

•مانتیس می تواند بین ۱ تا ۴ باشد، لذا باید در نهایت نرمال شود.

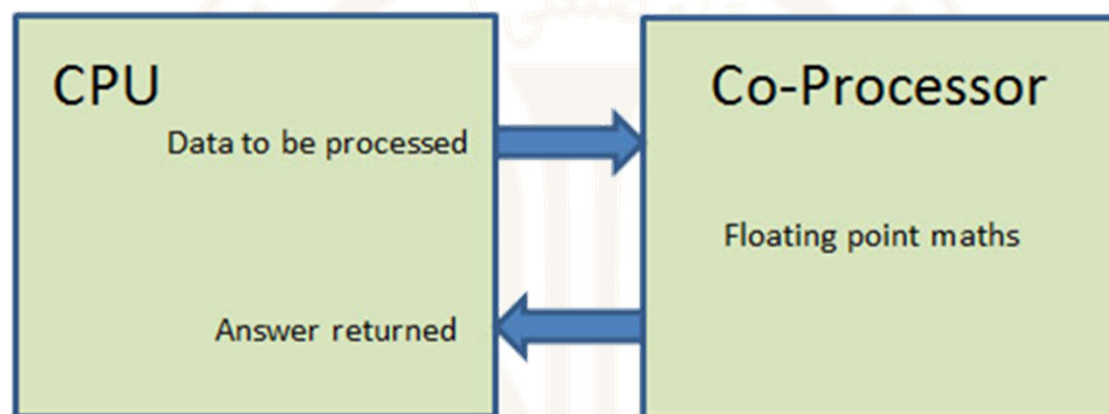
# توابع محاسباتی و مثلثاتی

• رجوع به جدول

• تخمین با استفاده از دنباله ها

• روش CORDIC

# پردازنده کمکی اعداد اعشاری



# پردازنده کمکی اعداد اعشاری (ادامه)

