

تفریق کننده ها

طراحی واحد منطق و حساب

Arithmetic logic unit (ALU) design

© تمامی اطلاعات موجود در این سند متعلق به دانشگاه صنعتی امیرکبیر بوده و حقوق قانونی آن محفوظ است.



عمل محاسباتی: تفریق

نوع نمایش: بی علامت

تفریق ۲ عدد n -بیتی:

- تفریق کننده آبشاری (Ripple subtractor)
- تفریق کننده مکمل گیر (Complement subtractor)



تفریق کننده اعداد بی علامت



عمل محاسباتی: تفریق

نوع نمایش: بی علامت

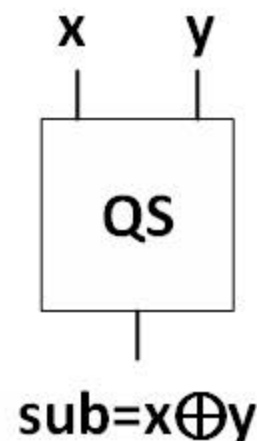
◀ $n=1$ (تفریق دو عدد بی علامت تک بیتی)

○ ربع تفریق کننده (Quarter subtractor)

○ نیم تفریق کننده (Half subtractor)

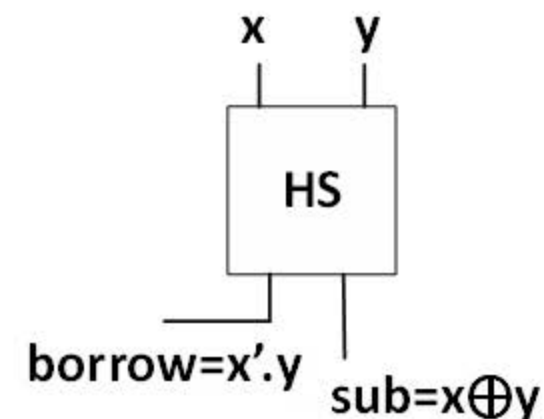
○ تمام تفریق کننده (Full subtractor)

x	y	s=x-y (تفریق ریاضی)
0	0	0
0	1	1
1	0	1
1	1	0



delay(sub) = d
Cost = 1 g

x	y	(تفریق ریاضی)	
		borrow	sub
0	0	0	0
0	1	1	1
1	0	0	1
1	1	0	0



delay(sub) = d
delay(borrow) = 2d
Cost = 3 g

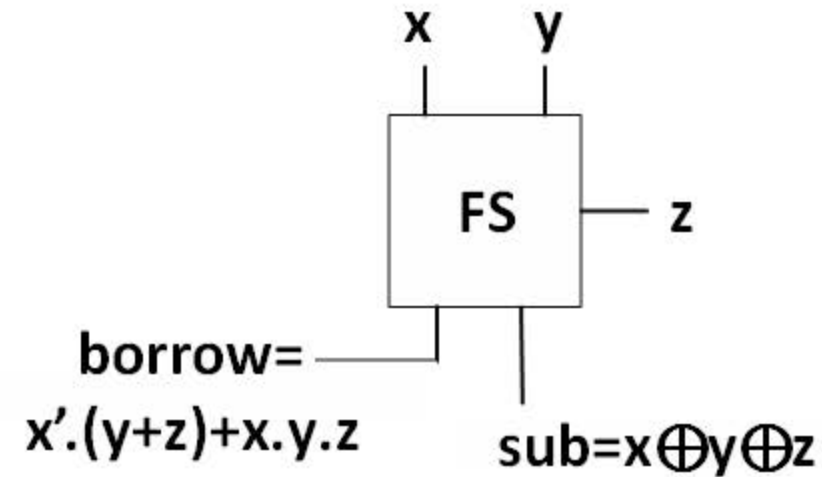


عمل محاسباتی: تفریق

نوع نمایش: بی علامت

$n=1$ (تفریق دو عدد بی علامت تک بیتی)
 ○ تمام تفریق کننده (Full subtractor)

x	y	z	x-y-z (تفریق ریاضی)	
			borrow	sub
0	0	0	0	0
0	0	1	1	1
0	1	0	1	1
0	1	1	1	0
1	0	0	0	1
1	0	1	0	0
1	1	0	0	0
1	1	1	1	1



$\text{delay (sub)} = d$
 $\text{delay (borrow)} = 3d$
 $\text{cost} = 6g$

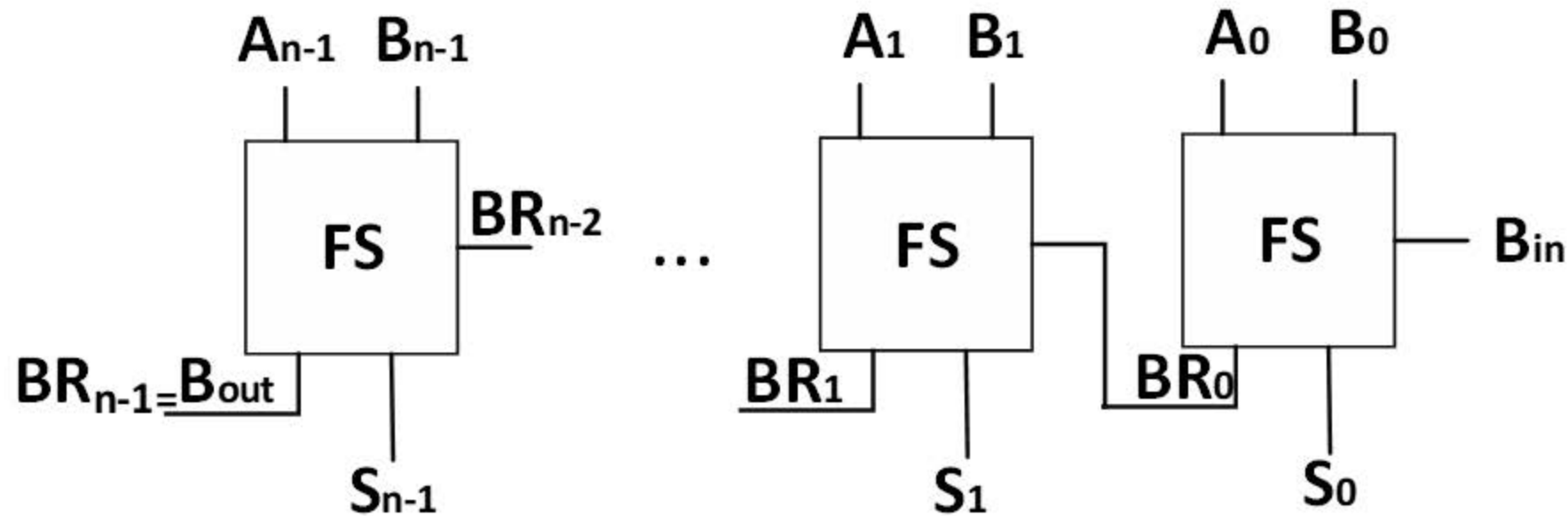
FS delay = 3d



تفریق کننده آبشاری (Ripple subtractor)



تفریق کننده آبشاری (Ripple subtractor)



$$\begin{aligned} \text{delay (subtract)} &= (3n-2)d \\ \text{delay (borrow)} &= 3nd \end{aligned}$$

$$\text{cost} = 6n g$$



تفریق کننده مکمل گیر (Complement subtractor)



تفریق کننده مکمل گیر

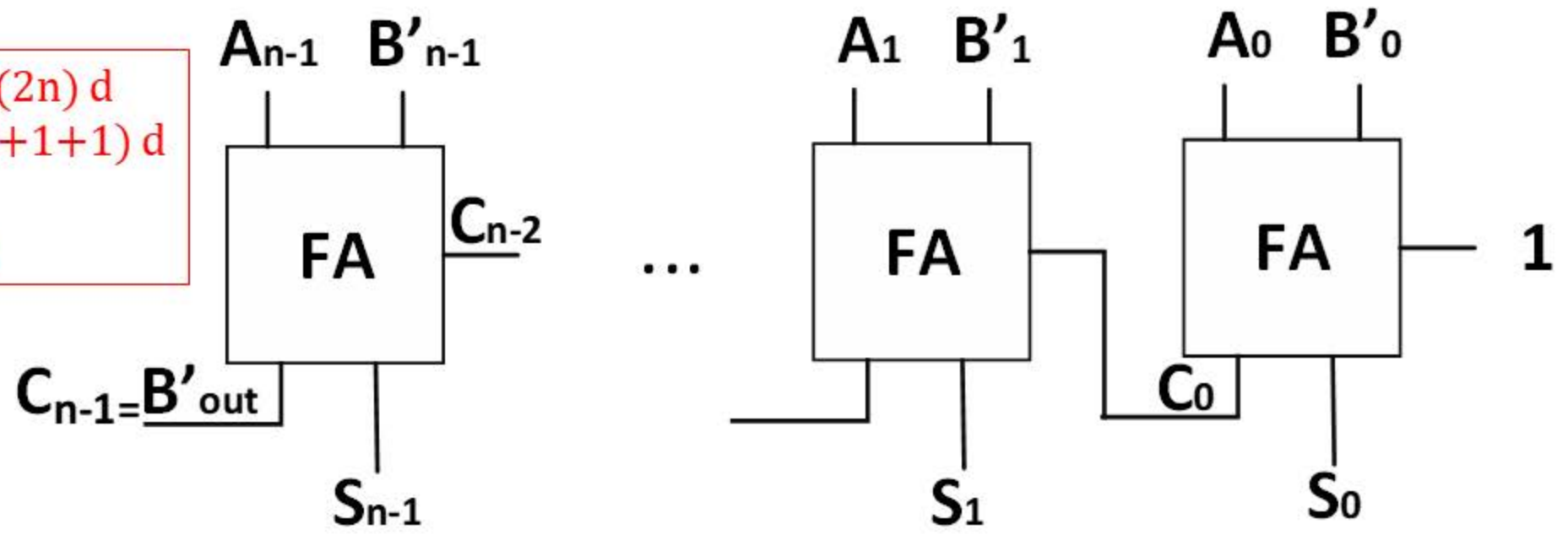
برای محاسبه $A - B$ در فضای n بیتی بصورت زیر عمل می کنیم:

$$\begin{aligned} A - B &= (\text{in } n \text{ bit}) = 2^n + A - B \\ &= A + (2^n - B) \\ &= (\text{provable}) = A + (B' + 1) \end{aligned}$$



تفریق کننده مکمل گیر

$\text{delay (subtract)} = (2n) d$
 $\text{delay (borrow)} = (2n+1+1) d$
 $\text{cost} = 6n g$



در این روش، رقم نقلی تولید شده، برعکس رقم قرضی است. چرا؟

$$\text{Bout} = \text{NOT} (C_{n-1})$$



سرریز یا Overflow

➤ در محاسبات **ALU** لازم است نتیجه محاسبات در n بیت قابل نمایش باشد (چرا؟)

➤ در محاسبات، گاهی اوقات شرایطی پیش می‌آید که پاسخ محاسبه دو عدد n -بیتی در فضای n -بیتی نادرست است (پاسخ غلط). برای اطلاع به کاربر، سیگنالی به نام **Overflow** (سرریز) در نظر گرفته می‌شود که از این طریق، اطلاع داده می‌شود. لذا قبل از برداشت پاسخ از مدار، لازم است ابتدا بیت سرریز چک شود که آیا یک است یا صفر.

○ اگر صفر باشد، پاسخ در n بیت، درست است و ارسال به مرحله بعدی است.

○ اگر یک باشد، پاسخ نادرست است و امکان انجام وجود ندارد. یعنی سخت افزار قادر به پاسخ دادن نیست (راه حل در این شرایط چیست؟)



شرایط سرریز شدن (پاسخ غلط) در محاسبات جمع و تفریق، بی علامت

◀ هنگام **جمع** دو عدد بی علامت

○ **ایجاد رقم نقلی**، به معنای سرریز شدن نتیجه است. (زیرا نتیجه $n+1$ رقمی است و قابل نمایش در n بیت نیست)

◀ هنگام **تفریق** دو عدد بی علامت

○ **ایجاد رقم قرضی**، به معنای سرریز شدن نتیجه است. (زیرا نتیجه منفی است و چون قابل نمایش نیست بیت قرضی ایجاد شده است.)

■ توجه: چنانچه از مدار تفریق گر مکمل گیر استفاده شود، نبود بیت نقلی، به معنای وجود رقم قرضی است. پس نبود بیت نقلی یعنی سرریز شدن نتیجه تفریق.

◀ مثال: مشخص کنید در فضای ۴-بیتی، کدام سرریز می شود و کدام سرریز نمی شود؟

0101	
0011	+

1011	
1101	+

0101	
0011	-

0011	
0101	-



سوال؟

