



سیستم‌های دیجیتال ۲

اسلاید ۱

انتقال ثبات‌ها و ریز عمل‌ها

محمد رضا رازقی‌زاده



رئوس مطالب

Register Transfer Language	زبان انتقال ثبات
Register Transfer	انتقال ثبات
Bus and Memory Transfers	انتقال‌های گذرگاهی و حافظه‌ای
Arithmetic Microoperations	ریزعمل‌های حسابی
Logic Microoperations	ریزعمل‌های منطقی
Shift Microoperations	ریزعمل‌های شیفت
Arithmetic, Logic, Shift Unit	واحد حساب، منطق و شیفت



سیستم دیجیتالی

➤ یک سیستم دیجیتال، مجموعه‌ای از ماژول‌های سخت افزاری متصل بهم است که کاری خاص را در زمینه پردازش اطلاعات انجام می‌دهند.

✓ در ساده‌ترین حالت طراحی سیستم‌های دیجیتال از مدارات ترتیبی و ترکیبی همچون ثبات‌ها، دیکدرها، عناصر حسابی و کنترل منطقی استفاده می‌شود.

➤ ساده‌ترین سیستم‌های دیجیتالی معمولاً با دو شاخصه قابل تعریف می‌باشند:

۱. ثبات‌های موجود در سیستم،

۲. عملیاتی که سیستم روی داده‌های ذخیره شده انجام می‌دهد.

➤ بنابراین، برای شناسایی يك سیستم باید بدانیم:

– چه عملیاتی روی داده‌ها انجام می‌شود.

– چه اطلاعاتی بین ثبات‌ها منتقل می‌شود.



ریزعمل

➤ عملیاتی که روی داده‌های ذخیره شده در ثبات‌ها انجام می‌شود **ریز عمل** نامیده شده و نتیجه آن در ثبات قبلی یا در ثباتی (ثبات‌های) دیگر ذخیره می‌شود.

➤ نمونه‌هایی از متداول‌ترین ریزعمل‌ها عبارتند از:

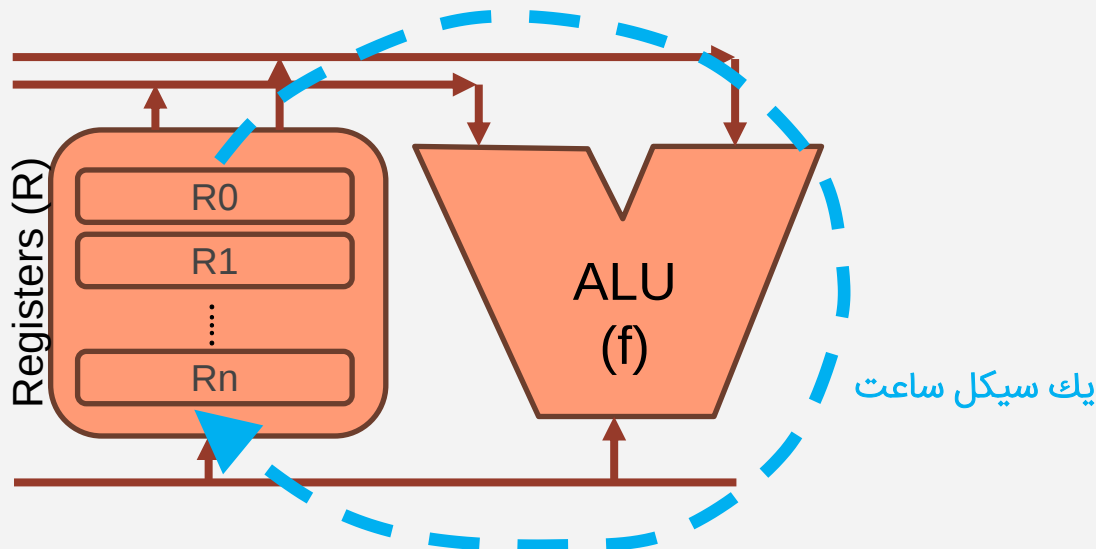
- پاک کردن Clear
- بار کردن Load
- شیفت Shift
- اضافه کردن Increment
- شمارش Count
- ...

➤ به عبارت دقیق‌تر، عملیات پایه که روی داده‌های ذخیره شده در يك یا چند ثبات در طی يك پالس ساعت انجام می‌شود **ریزعمل** نام دارد.



مثال:

$$R \leftarrow f(R, R)$$



تابع f می‌تواند هریک از ریز عمل‌های زیر باشد:

shift, load, clear, increment, add, subtract, and, xor, complement



➤ **سازمان سخت افزاری** داخلی کامپیوتر با سه شاخصه قابل تعریف می باشد:

- I. مجموعه ثبات های آن و وظیفه هر يك از آنها،
- II. مجموعه ریزعمل هایی که روی ثبات ها انجام می شود،
- III. سیگنال های کنترلی که ترتیب اجرای ریزعمل ها را مشخص می کنند (واحد کنترلی که موجب آغاز و اجرای رشته ریزعمل ها می شود)

➤ **بررسی کامپیوتر از این دیدگاه سطح انتقال ثبات (RTL) نامیده می شود.**

Register Transfer Level

✓ در این سطح تمرکز بر موارد زیر است:

- ثبات های سیستم
- تبدیل داده ها درون ثبات ها
- انتقال داده ها بین ثبات ها



➤ به جای تشریح لفظی رشته ریز عمل‌ها در يك سیستم، آن را به صورت سمبولیک نشان می دهند. این بیان سمبولیک، زبان انتقال ثبات نامیده می شود.

Register Transfer Language

➤ ویژگی‌های زبان انتقال ثبات:

- يك زبان سمبوليك است.
- يك ابزار آسان برای شرح سازمان داخلی کامپیوترهای دیجیتال است.
- فرآیند طراحی سیستم های دیجیتال را تسهیل می کند.



شرایط نامگذاری ثبات‌ها در سیستم‌های دیجیتال

- ثبات‌ها معمولاً با حروف بزرگ انگلیسی نامگذاری می‌شوند.
 - گاهی اوقات پس از اسم آنها اعدادی قرار می‌گیرد (A, R15, IR).
 - ثبات‌هایی که با نام R شروع می‌شوند، معمولاً ثبات‌هایی با کاربری عمومی هستند.
 - اغلب نام‌ها نشان دهنده کاری است که ثبات انجام می‌شود مثلاً:
- MAR - Memory Address Register
 - PC - Program Counter
 - IR - Instruction Register



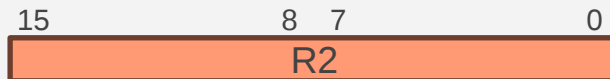
نمایش ثبات‌ها در سیستم‌های دیجیتال

Register



Numbering of bits

Showing individual bits



Subfields



Register Transfer

➤ کپی شدن اطلاعات يك ثبات به ثبات ديگر انتقال ثبات نام دارد.

➤ انتقال ثبات به شكل سمبوليك زير نشان داده مي شود:

$$R_2 \leftarrow R_1$$

اين نوشتار بيانگر اين است كه:

– در اين حالت محتوای ثبات R_1 به R_2 منتقل می شود.

– انتقال در يك پالس انجام می شود.

– محتوای R_1 تغيير نمی کند.



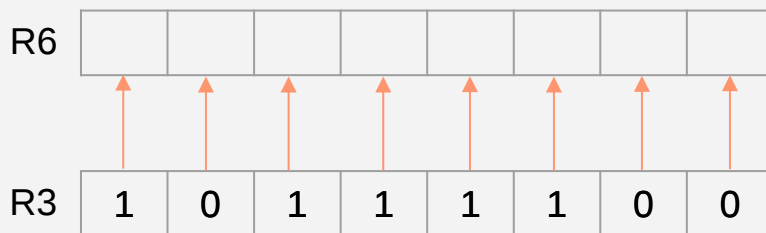
مثال:

يك انتقال ثبات مثل زیر:

$R6 \leftarrow R4$

موارد زیر را در سیستم ایجاب می کند:

- خطوط انتقال از $R4$ به $R6$
- بار شدن موازی در $R6$
- خطوط کنترل لازم برای انجام عملیات



Start



- در بسیاری مسائل در حوزه‌های مختلف علوم، عملیات فقط زمانی که يك شرط خاص برقرار باشد باید اجرا شوند.
- این مسأله شبیه if در زبان های برنامه نویسی است.
- در سیستم های دیجیتال پیاده‌سازی شرط با يك **سیگنال کنترلی** یا **تابع کنترلی** انجام می شود.

P: $R2 \leftarrow R1$

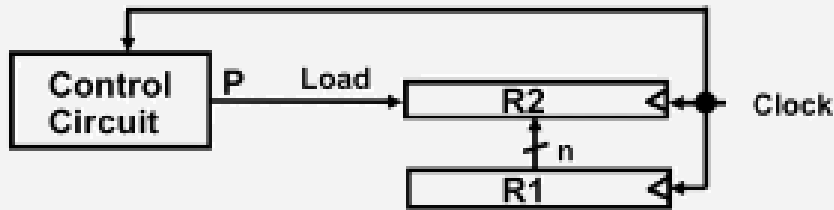
عبارت فوق در زبان انتقال ثبات‌ها بیانگر این است که اگر P برابر ۱ بود انتقال از R۱ به R۲ انجام شود، به بیان دیگر:

if (P = 1) then ($R2 \leftarrow R1$)

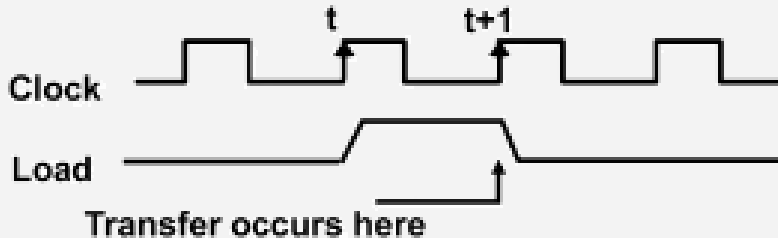


پیاده سازی سخت افزاری انتقالات کنترلی

P: $R2 \leftarrow R1$



بلوک دیاگرام



دیاگرام زمان بندی

فرض می شود که ثبات ها حساس به لبه مثبت هستند



➤ اگر تعداد دو یا بیشتر **عملیات همزمان** انجام شود آنها را با کاما، از هم جدا می کنیم.

P: $R3 \leftarrow R5, MAR \leftarrow IR$

در این عبارت اگر $P=1$ باشد، به طور همزمان $R5$ به $R3$ و IR به MAR منتقل می شود.

➤ اگر برای انجام یک عبارت چندین شرط به صورت همزمان برقرار باشد، داریم:

$X1 + X2 : R3 \leftarrow R2$

در این عبارت اگر $X1$ یا $X2$ برابر ۱ باشند $R2$ به $R3$ منتقل می شود، به عبارت دیگر:

If $X1 = 1$ or $X2 = 1$ then $R3 \leftarrow R2$



مثال

$X_1 + X_2 : R3 \leftarrow R1 + R2$

این عبارت یعنی:

If $X_1 = 1$ or $X_2 = 1 : R3 \leftarrow R1 + R2$

مثال

$X_1 + X_2 : R3 \leftarrow R1 \vee R2$

این عبارت یعنی:

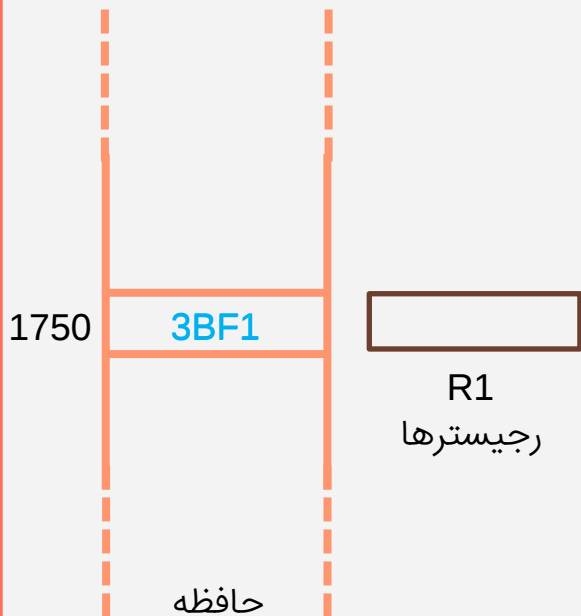
If $X_1 = 1$ or $X_2 = 1 : R3 \leftarrow R1$ or $R2$

مثال

$X_1 + X_2 : R3 \leftarrow R1 \wedge R2$

این عبارت یعنی:

If $X_1 = 1$ or $X_2 = 1 : R3 \leftarrow R1$ and $R2$



مثال

$\bar{X} : R1 \leftarrow M [AR]$

در این عبارت، به شرطی که سیگنال کنترلی X برابر با صفر شود، محتوای سطری از حافظه که ثبات آدرس (AR) به آن اشاره می کند در ثبات R1 قرار می گیرد.

مثال

$\bar{X} : R1 \leftarrow M [1750]$

در این عبارت، به شرطی که سیگنال کنترلی X برابر با صفر شود، محتوای سطر ۱۷۵۰ از حافظه در ثبات R1 قرار می گیرد.



نمادهای اساسی در انتقال ثبات‌ها

مثال	شرح	نماد
MAR , R _Y	نشان دهنده يك ثبات	حروف بزرگ
R _Y (o-Y) , R _Y (L)	نشان دهنده قسمتی از يك ثبات	پرانتز ()
R _Y ← R _I	نشان دهنده انتقال اطلاعات	پیکان ←
P:	نشان دهنده پایان تابع کنترلی	دو نقطه :
A←B , B←A	جداکننده دو ریزعمل	کاما ،



ارتباط بین ثبات ها

➤ یک کامپیوتر دیجیتال نوعی شامل تعداد زیادی ثبات شده و باید در آن مسیرهایی برای انتقال اطلاعات از یک ثبات به ثبات دیگر فراهم شود. اما، اتصال مستقیم هر ثبات با ثبات دیگر به دلیل ایجاد پیچیدگی زیاد در سیم بندی امکان پذیر نیست.

✓ برای اتصال n ثبات به یکدیگر به $n(n-1)$ خط ارتباطی نیاز است.

✓ هزینه: $O(n^2)$

برای سیستم هایی با تعداد ثبات زیاد، عملی نیست



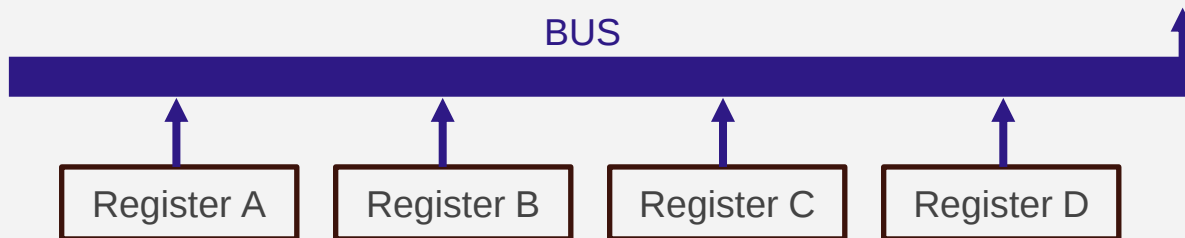


➤ از آنجا که ارتباط دو به دو رجیسترها با پیچیدگی بسیار در سیم بندی همراه می باشد، بهتر است که از يك مسیر ارتباطی مشترك به نام **BUS** یا **گذرگاه مشترك** استفاده شود. گذرگاه يك مسیر (شامل گروهی از سیم ها) است که اطلاعات روی آن منتقل می شود. انتقال می تواند از منابع مختلف به مقاصد مختلف باشد.

✓ گذرگاه مشترك شامل مداراتی متمرکز برای ایجاد مسیر انتقال داده ها می باشد.

✓ گذرگاه مشترك شامل مداراتی كنترلی برای انتخاب ثبات مبداء و ثبات مقصد است.

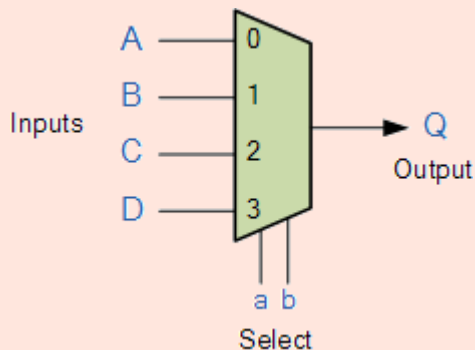
$BUS \leftarrow R$





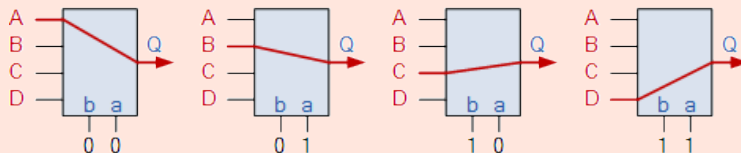
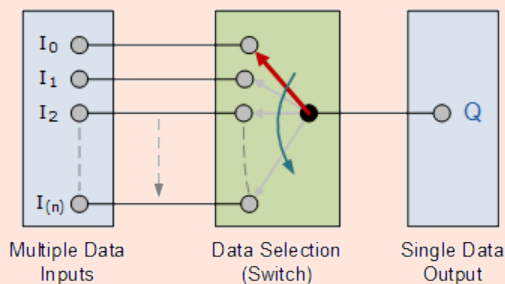
گذرگاه مشترک به کمک مالتی پلکسر (MUX)

یادآوری



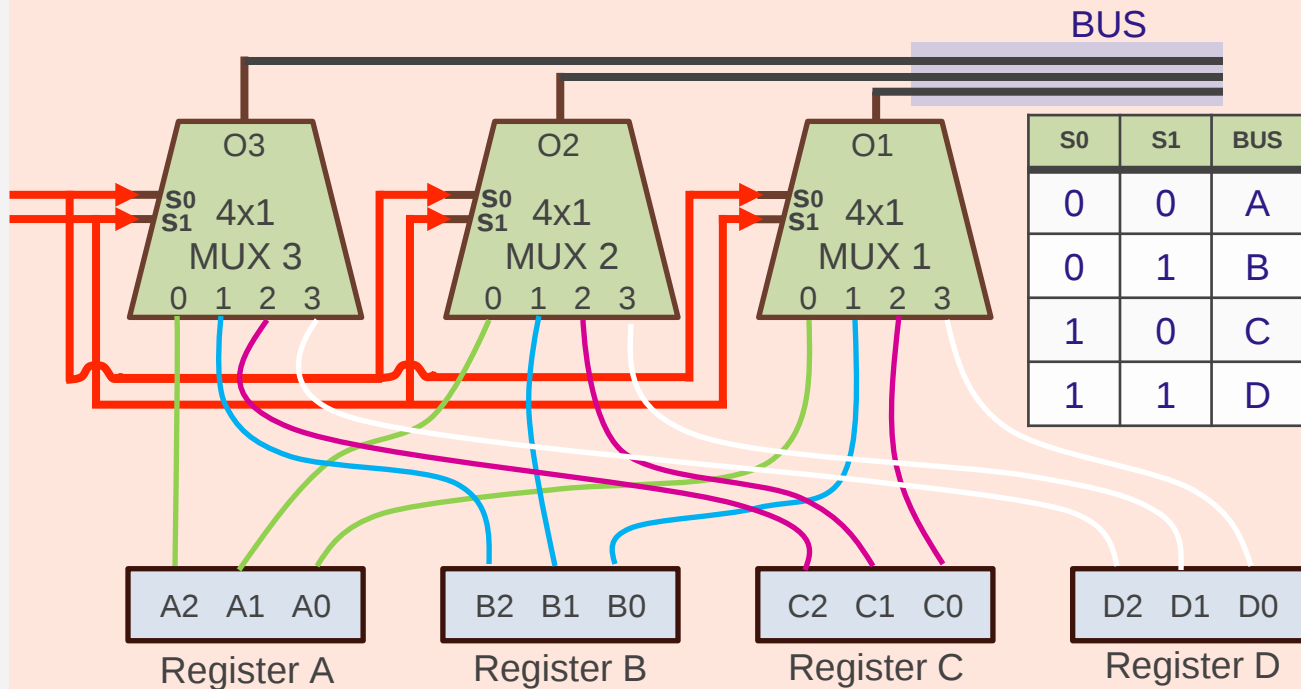
Truth Table

Select		Inputs					Q
b	a	D	C	B	A		
0	0	x	x	x	1	1	
0	1	x	x	1	x	1	
1	0	x	1	x	x	1	
1	1	1	x	x	x	1	



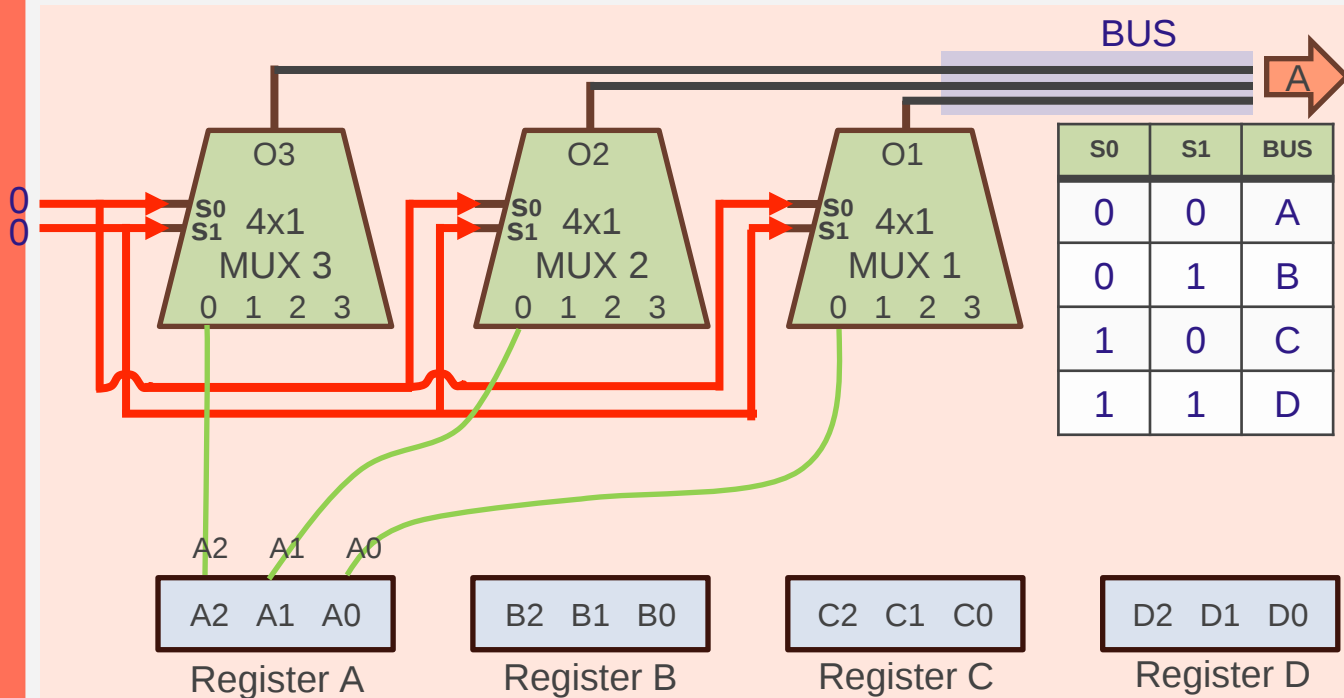


گذرگاه مشترک به کمک مالتی پلکسر (MUX)



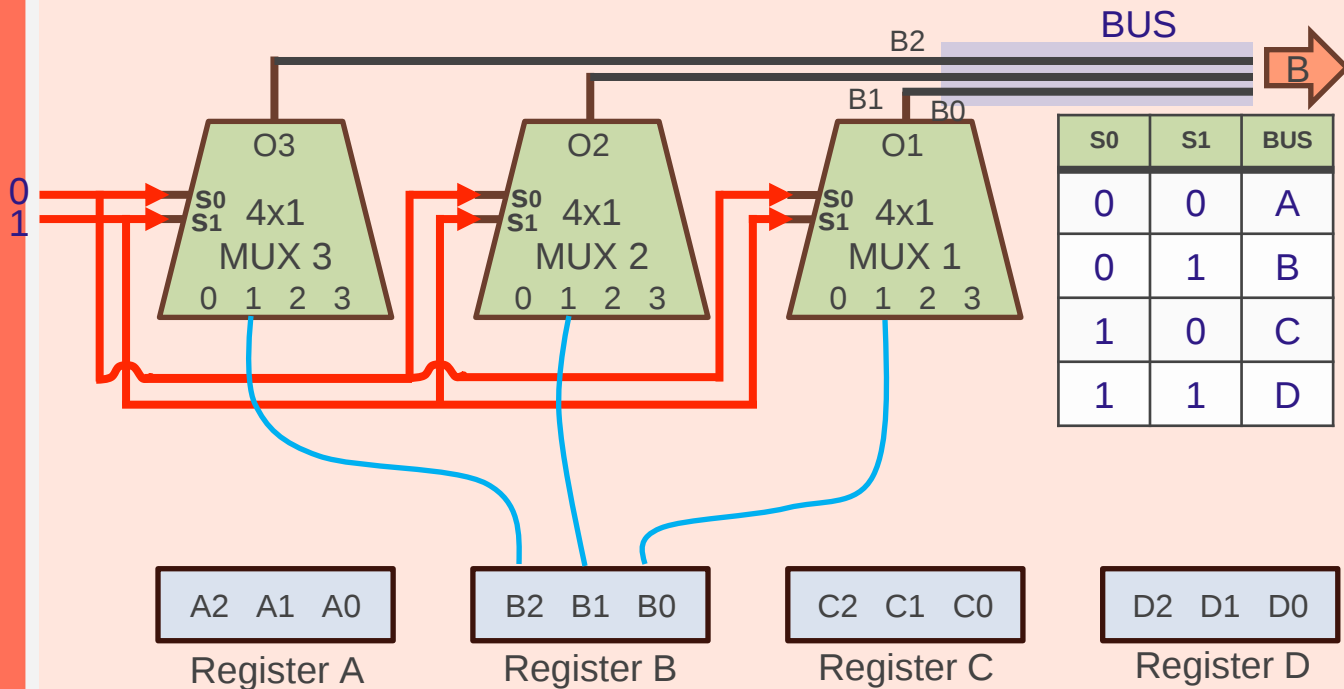


گذرگاه مشترک به کمک مالتی پلکسر (MUX)





گذرگاه مشترک به کمک مالتی پلکسر (MUX)





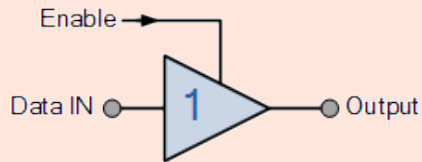
گذرگاه مشترک به کمک مالتی پلکسر (MUX)

- ✓ اگر k ثبات n بیتی داشته باشیم، برای ایجاد گذرگاه مشترک به n مالتی پلکسر $k \times 1$ نیاز داریم.
- ✓ تعداد بیت‌های ثبات‌ها با تعداد MUX‌ها برابر است.

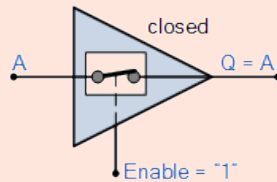
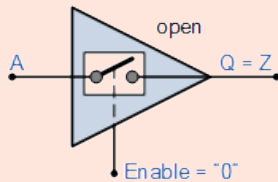


گذرگاه مشترك به كمك بافر سه حالته

يادآوری



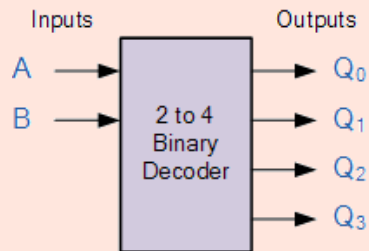
Enable	IN	OUT
0	0	Hi-Z
0	1	Hi-Z
1	0	0
1	1	1





گذرگاه مشترك به كمك بافر سه حالته

يادآوري

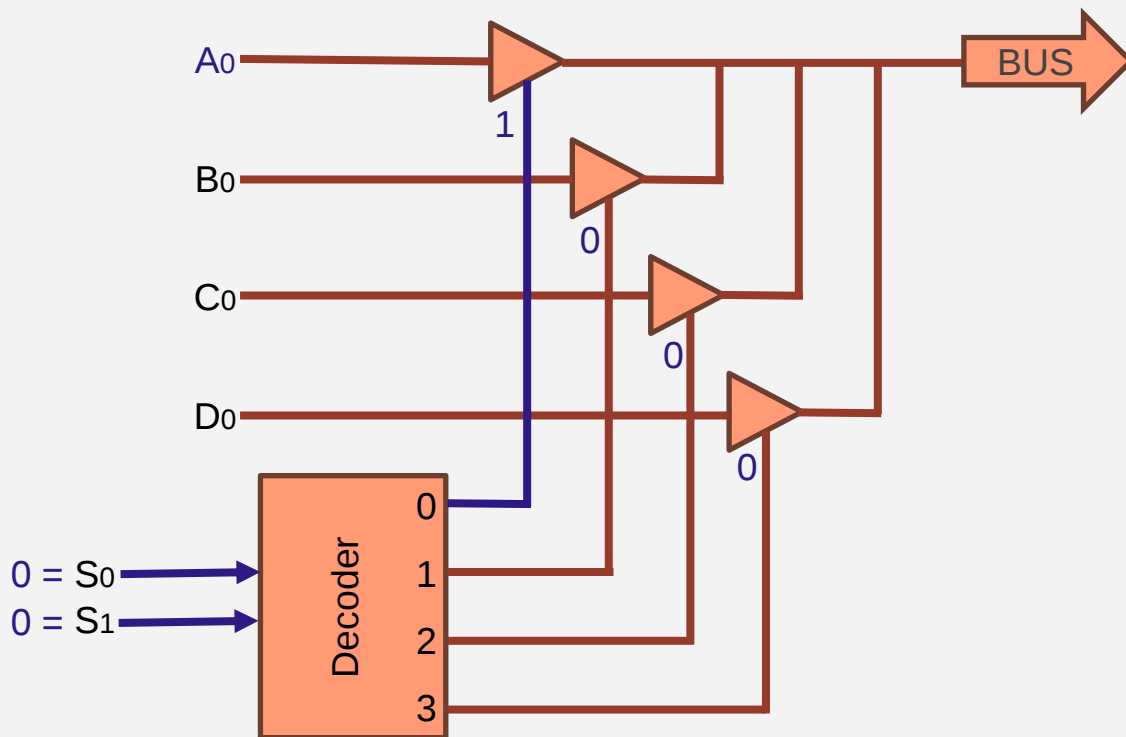


Truth Table

A	B	Q ₀	Q ₁	Q ₂	Q ₃
0	0	1	0	0	0
0	1	0	1	0	0
1	0	0	0	1	0
1	1	0	0	0	1

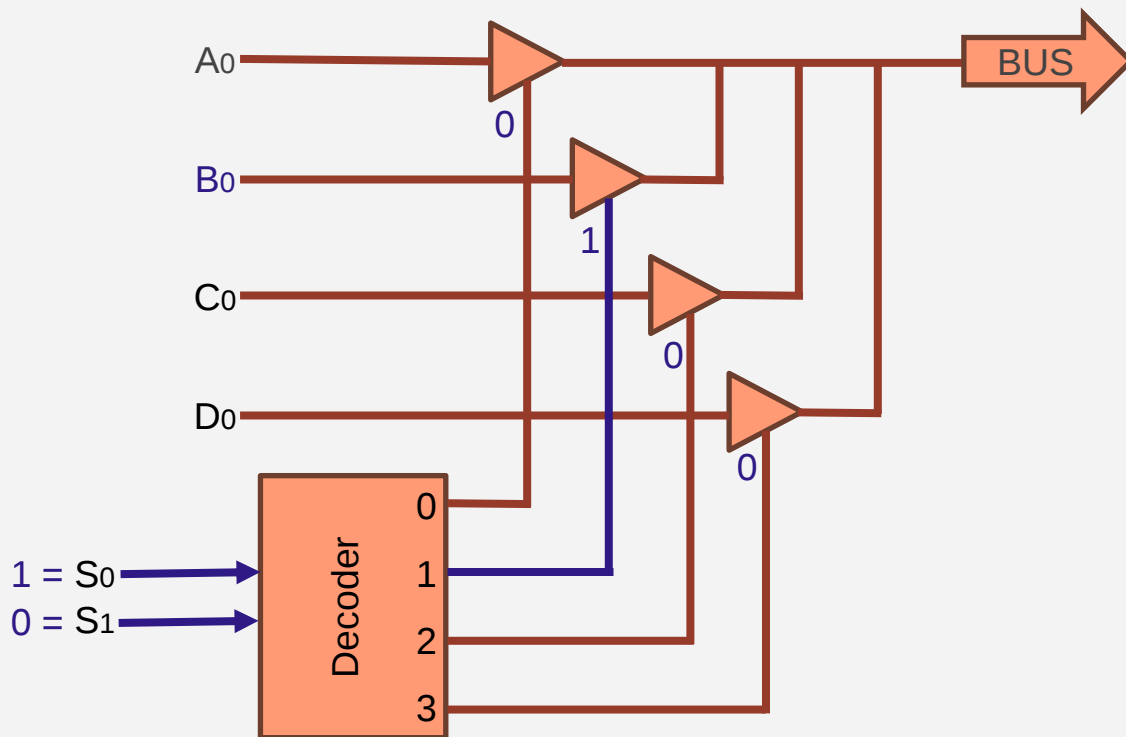


گذرگاه مشترك به كمك بافر سه حالته



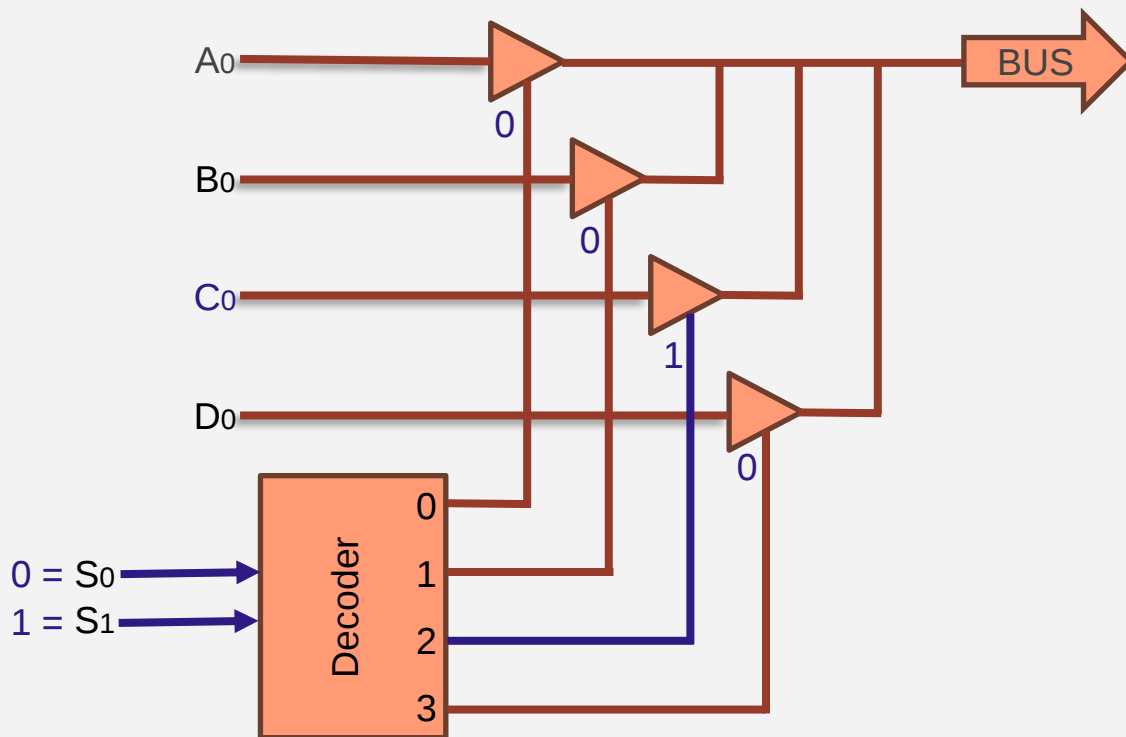


گذرگاه مشترك به كمك بافر سه حالته



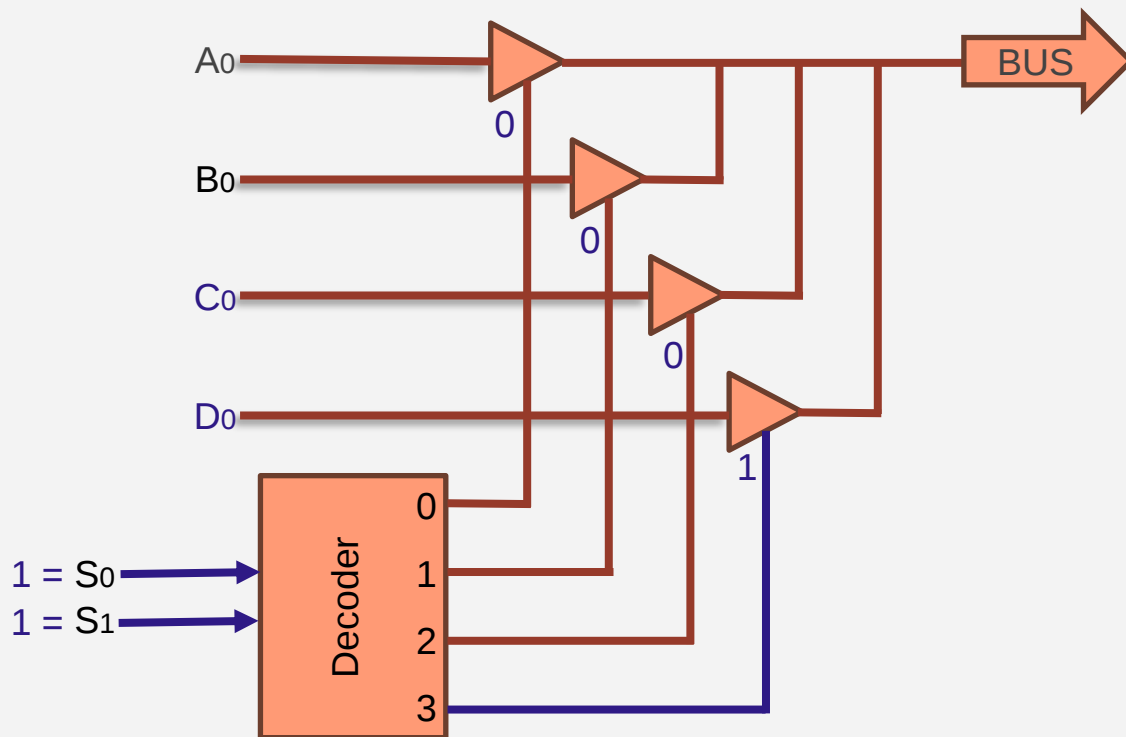


گذرگاه مشترك به كمك بافر سه حالته





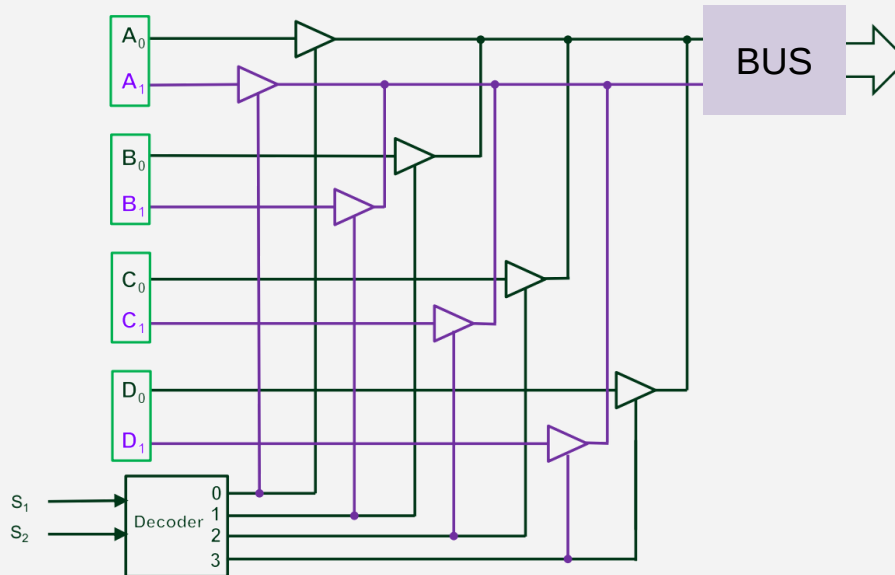
گذرگاه مشترك به كمك بافر سه حالته





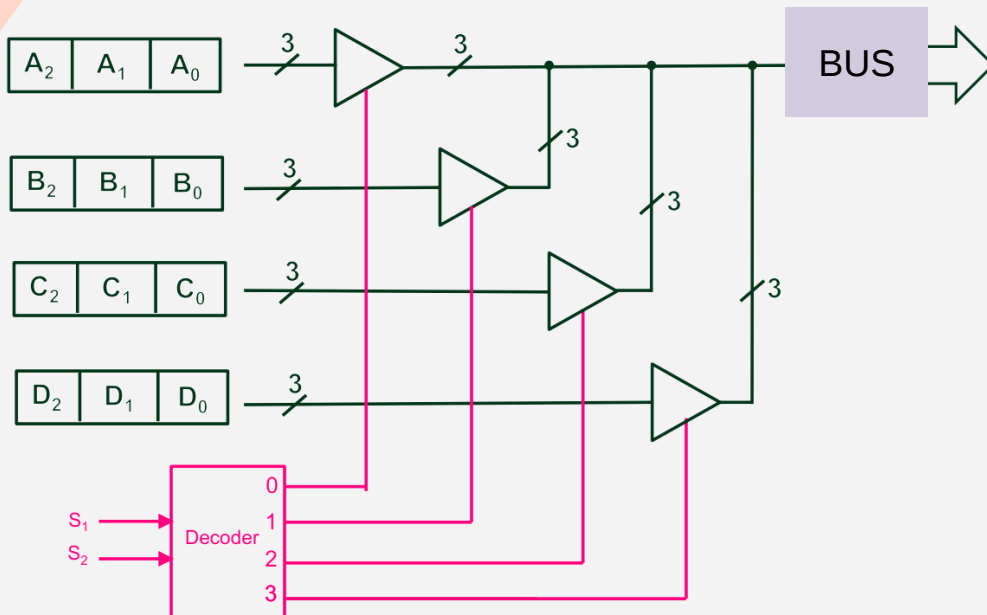
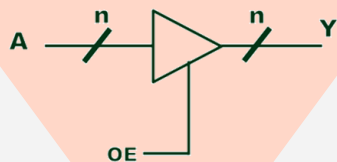
گذرگاه مشترك به كمك بافر سه حالته

✓ اگر k ثبات n بیتی داشته باشیم، برای ایجاد گذرگاه مشترك به كمك بافر سه حالته به يك ديكر k خروجی و $n \times k$ بافر سه حالته نیاز داریم.





گذرگاه مشترك به كمك بافر سه حالته





نمایش انتقال گذرگاه به بیان RTL

➤ انتقال ثبات از طریق گذرگاه می تواند به یکی از دو شکل زیر نشان داده شود.

$$R_2 \leftarrow R_1$$

یا

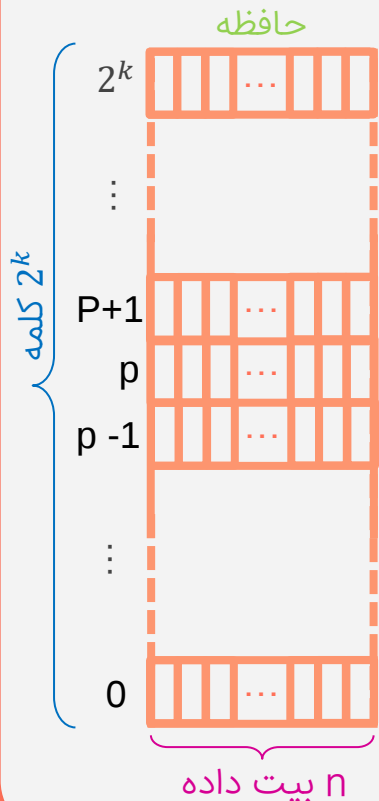
$$BUS \leftarrow R_1, R_2 \leftarrow BUS$$

➤ در اولی گذرگاه به صورت ضمنی وجود دارد درحالی که در دومی به طور صریح بیان شده است.



حافظه (RAM)

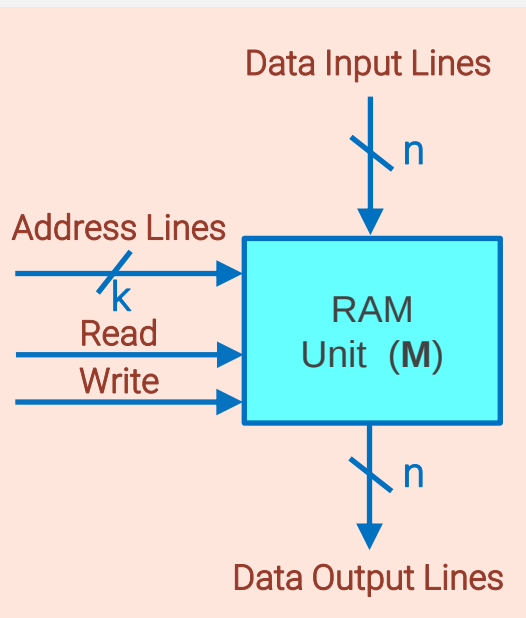
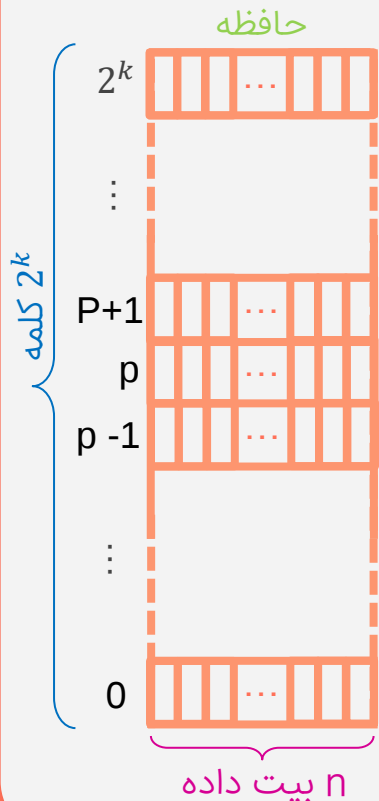
- ✓ حافظه را می‌توان به صورت یک مدار ترتیبی در نظر گرفت که شامل تعدادی ثبات می‌باشد.
- ✓ این ثبات‌ها کلمات حافظه را در خود نگهداری می‌کنند.
- ✓ هر کلمه با يك آدرس مشخص می‌شود.
- ✓ گستره آدرس‌ها از ۰ تا $2^k - 1$ وسعت خواهد داشت.
- ✓ در واقع برای $r = 2^k$ کلمه آدرس دهی از ۰ تا $2^k - 1$ می‌باشد.
- ✓ هر کلمه (ثبات) می‌تواند n بیت را ذخیره کند.





حافظه (RAM)

يك RAM با $r = 2^k$ كلمه را در نظر بگيريد. اين RAM به موارد
زير نياز دارد:



- n خط داده ورودی
- n خط داده خروجی
- k خط آدرس
- يك خط کنترل خواندن
- يك خط کنترل نوشتن



مدل حافظه (حافظه با دستیابی تصادفی موسوم به ram)

• حافظه را می‌توان به صورت یک آرایه بزرگ از N عدد صحیح در نظر گرفت، که بوسیله اندیس قابل دستیابی می‌باشد.

N-1	0
.	.
.	.
.	.
2	56
1	4
0	18

↑
آدرس

↑
محتوی

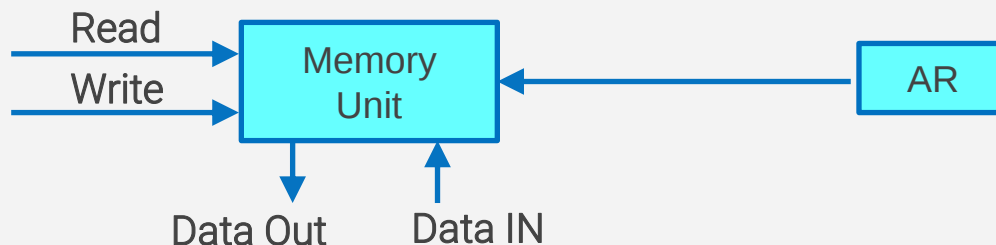
✓ به عنوان نمونه ، $M[1]$ شامل مقدار ۴ است.

✓ این مکان‌ها صرفاً در دسترس است و می‌توان در آن‌ها نوشتن و خواندن را انجام داد.



انتقال حافظه

- ✓ در سطح انتقال ثبات، حافظه با نماد M نشان داده می‌شود.
- ✓ از آنجاکه حافظه شامل چندین آدرس مختلف است، باید آدرس مورد نظر در حافظه مشخص شود.
- ✓ در سیستم‌های کامپیوتری برای دستیابی به حافظه، آدرس دلخواه در یک ثبات مشخص قرار داده می‌شود که این ثبات **Memory Address Register (MAR)** یا **(AR)** نامیده می‌شود.
- ✓ وقتی حافظه دستیابی می‌شود، محتوای MAR به عنوان آدرس روی خطوط آدرس حافظه مورد استفاده قرار می‌گیرد.





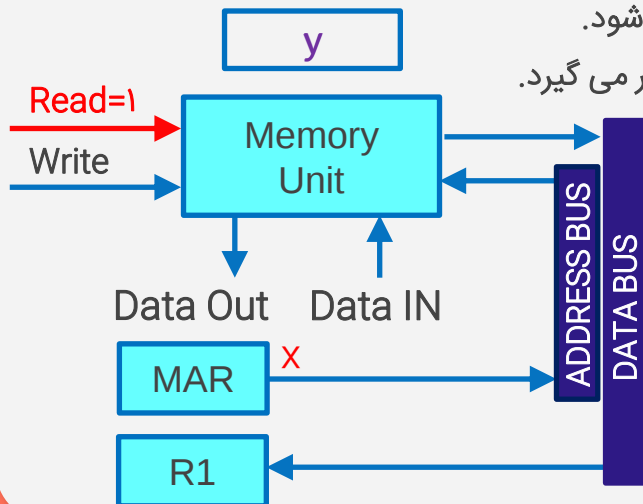
خواندن از حافظه

برای خواندن يك کلمه از حافظه زبان انتقال ثبات بصورت زیر است:

$$R1 \leftarrow M[MAR]$$

برای انجام مثال فوق اعمال زیر انجام می پذیرد:

- محتوای MAR روی خطوط آدرس فرستاده می شود.
- سیگنال (= ۱) Read به واحد حافظه فرستاده می شود.
- محتوای آدرس مشخص شده روی خطوط داده قرار می گیرد.
- این مقدار از گذرگاه به ثبات R1 منتقل می شود.





نوشتن در حافظه

برای نوشتن در يك کلمه حافظه زبان انتقال ثبات بصورت زیر نوشته می شود:

$$M[MAR] \leftarrow R1$$

برای انجام مثال فوق اعمال زیر انجام می پذیرد:

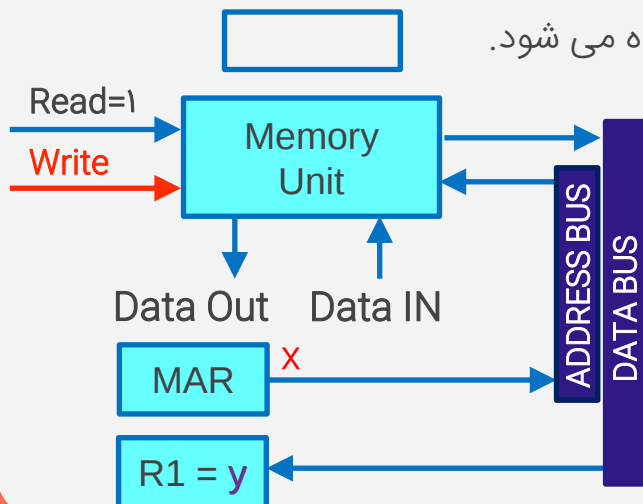
– محتوای MAR روی خطوط آدرس فرستاده می شود.

– سیگنال (= ۱) write به واحد حافظه فرستاده می شود.

– مقدار ثبات R1 به گذرگاه منتقل می شود.

– مقدار روی باس به محل مشخص شده در

حافظه منتقل می شود.





خلاصه ای از ریزعمل های انتقال ثبات

$A \leftarrow B$	Transfer content of reg. B into reg. A
$AR \leftarrow DR(AD)$	Transfer content of AD portion of reg. DR into reg. AR
$A \leftarrow \text{constant}$	Transfer a binary constant into reg. A
$ABUS \leftarrow R1$	Transfer content of R1 into bus A and. at the same time.
$R2 \leftarrow ABUS$	Transfer content of bus A into R2
AR	Address register
DR	Data register
$M[R]$	Memory word specified by reg. R
M	Equivalent to $M[AR]$
$DR \leftarrow M$	Memory read operation: transfers content of Memory word specified by AR into DR
$M \leftarrow DR$	Memory write operation: transfers content of DR into memory word specified by AR



ریزعمل ها در سیستم کامپیوتری

- ۱ - ریزعمل های انتقال ثبات (Register Transfer Microoperations)
- ۲ - ریزعمل های حسابی (Arithmetic Microoperations)
- ۳ - ریزعمل های منطقی (Logic Microoperations)
- ۴ - ریزعمل های شیفت (Shift Microoperations)



ریزعمل های حسابی پایه عبارت اند:

✓ افزایش يك واحد

✓ جمع

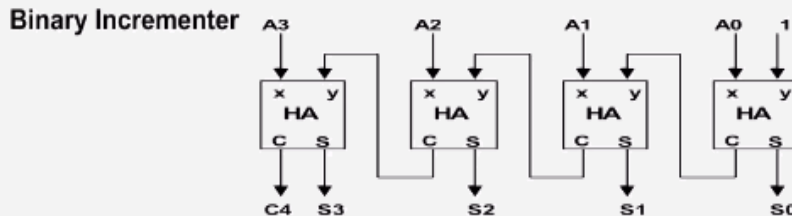
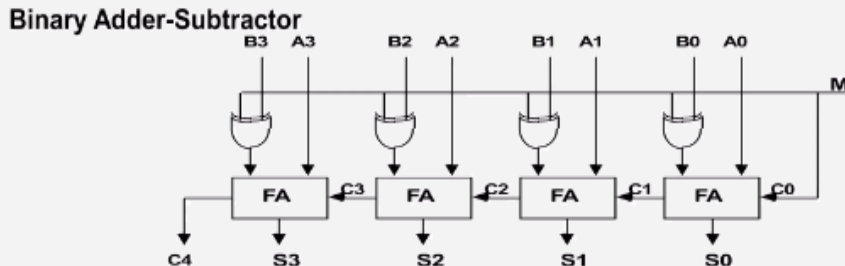
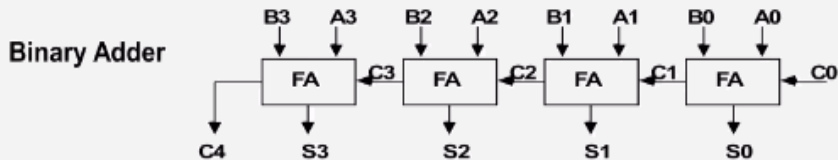
✓ کاهش يك واحد

✓ تفریق

$R3 \leftarrow R1 + R2$	Contents of R1 plus R2 transferred to R3
$R3 \leftarrow R1 - R2$	Contents of R1 minus R2 transferred to R3
$R2 \leftarrow R2'$	Complement the contents of R2
$R2 \leftarrow R2' + 1$	2's complement the contents of R2 (negative number)
$R3 \leftarrow R1 + R2' + 1$	Subtraction
$R1 \leftarrow R1 + 1$	Increment
$R1 \leftarrow R1 - 1$	Decrement

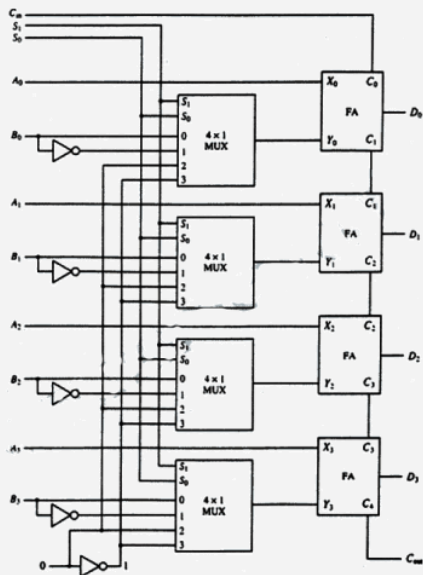


افزایشگر، جمع کننده و تفریق کننده دودویی





مدار عملیات حسابی



Si	So	Cin	Y	Output	ریز عمل‌ها	
0	0	0	B	$D=A+B$	جمع	Add
0	0	1	B	$D=A+B+1$	جمع با بیت انتقال	Add with carry
0	1	0	B'	$D=A+B'$	تفریق با بیت قرضی	Subtract with borrow
0	1	1	B'	$D=A+B'+1$	تفریق	Subtract
1	0	0	0	$D=A$	انتقال A	Transfer A
1	0	1	0	$D=A+1$	A افزایش	Increment A
1	1	0	1	$D=A-1$	A کاهش	Decrement A
1	1	1	1	$D=A$	انتقال A	Transfer A



- ✓ ریز عمل‌هایی هستند که عملیات دودویی را روی رشته‌ای از بیت‌های ثبات انجام می‌دهند.
- ✓ عملیات منطقی روی يك بیت داده کار می‌کنند به همین دلیل به آنها bit-wise گویند. به طور مثال، در يك ثبات هشت بیتی وقتی عمل not انجام می‌شود روی هر بیت به طور مستقل انجام می‌شود.

➤ به طور کلی ۱۶ عملیات متفاوت منطقی می تواند روی دو متغیر دودویی انجام شود:

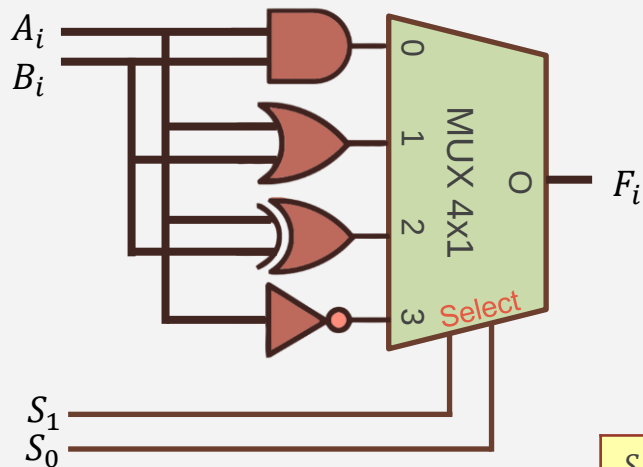
[illegible]



لیست ریزعمل های منطقی

- ✓ با n متغیر می توان 2^{2^n} عمل منطقی مختلف روی 2 متغیر را تعریف نمود.
- ✓ جدول ارزش برای دو متغیر دودویی

X y	0 0 1 1 0 1 0 1	Boolean Faunction	Micro- Operations	Name
	0 0 0 0	$F0 = 0$	$F \leftarrow 0$	Clear
	0 0 0 1	$F1 = xy$	$F \leftarrow A \wedge B$	AND
	0 0 1 0	$F2 = xy'$	$F \leftarrow A \wedge B'$	
	0 0 1 1	$F3 = x$	$F \leftarrow A$	Transfer A
	0 1 0 0	$F4 = x'y$	$F \leftarrow A' \wedge B$	
	0 1 0 1	$F5 = y$	$F \leftarrow B$	Transfer B
	0 1 1 0	$F6 = x \oplus y$	$F \leftarrow A \oplus B$	Exclusive-OR
	0 1 1 1	$F7 = x + y$	$F \leftarrow A \vee B$	OR
	1 0 0 0	$F8 = (x + y)'$	$F \leftarrow (A \vee B)'$	NOR
	1 0 0 1	$F9 = (x \oplus y)'$	$F \leftarrow (A \oplus B)'$	Exclusive-NOR
	1 0 1 0	$F10 = y'$	$F \leftarrow B'$	Complement B
	1 0 1 1	$F11 = x + y'$	$F \leftarrow A \vee B'$	
	1 1 0 0	$F12 = x'$	$F \leftarrow A'$	Complement A
	1 1 0 1	$F13 = x' + y$	$F \leftarrow A' \vee B$	
	1 1 1 0	$F14 = (xy)'$	$F \leftarrow (A \wedge B)'$	NAND
	1 1 1 1	$F15 = 1$	$F \leftarrow \text{All } 1\text{'s}$	Set to All 1's



پیاده سازی سخت افزاری عملیات منطقی

✓ بیشتر سیستمها فقط چهار عمل
AND ($\wedge$), OR ($\vee$), XOR ($\oplus$),
Complement/NOT را پیاده سازی
کرده و سایر عملیات را به کمک این
چهار ریز عمل پیاده می کنند.

S_1	S_0	Output	μ - Operation
0	0	$F = A \wedge B$	AND
0	1	$F = A \vee B$	OR
1	0	$F = A \oplus B$	XOR
1	1	$F = A'$	Complement



کاربردهای ریزعمل‌های منطقی

- ✓ ریزعمل‌های منطقی می‌توانند برای دستکاری بیتی (تغییر بیت های يك قسمت دلخواه از يك ثبات) مورد استفاده قرار گیرند.
- ✓ فرض کنید داده ها در ثبات A هستند. ثبات B می‌تواند برای تغییر محتویات A به کار رود.

Selective-set	$A \leftarrow A + B$
Selective-complement	$A \leftarrow A \oplus B$
Selective-clear	$A \leftarrow A \cdot B'$
Mask (Delete)	$A \leftarrow A \cdot B$
Clear	$A \leftarrow A \oplus B$
Insert	$A \leftarrow (A \cdot B) + C$
Compare	$A \leftarrow A \oplus B$

...



ست کردن انتخابی

✓ در يك کردن انتخابی B برای تعیین بیت‌هایی از A که قرار است يك شوند مورد استفاده قرار می‌گیرد.

1 1 0 0	A_t	
1 0 1 0	B	
1 1 1 0	A_{t+1}	$(A \leftarrow A + B)$

OR Truth Table

A	B	Y
0	0	0
0	1	1
1	0	1
1	1	1

✓ به ازای بیت‌هایی که در B مقدار يك دارند، بیت‌های معادل آنها در A يك می‌شود.
بقیه بیت‌های A بدون تغییر می‌مانند.



مکمل کردن انتخابی

✓ در مکمل کردن انتخابی ثبات B برای تعیین بیت هایی از A که قرار است مکمل شوند مورد استفاده قرار می گیرد

1 1 0 0	A_t	
1 0 1 0	B	
0 1 1 0	A_{t+1}	$(A \leftarrow A \oplus B)$

XOR Truth Table

A	B	Y
0	0	0
0	1	1
1	0	1
1	1	0

✓ به ازای بیت هایی که در B مقدار یک دارند، بیت های معادل آنها در A مکمل (NOT) می شود. بقیه بیت های A بدون تغییر می مانند.



پاک کردن انتخابی

✓ در پاک کردن انتخابی ثابت B برای تعیین بیت هایی از A که قرار است پاک (صفر) شوند مورد استفاده قرار می گیرد

1 1 0 0	A_t	
1 0 1 0	B	
0 1 0 0	A_{t+1}	$(A \leftarrow A \cdot B')$

AND Truth Table

A	B	Y
0	0	0
0	1	0
1	0	0
1	1	1

✓ به ازای بیت هایی که در B مقدار يك دارند، بیت های معادل آنها در A صفر می شود.
بقیه بیت های A بدون تغییر می مانند.



عملیات ماسک کردن

- ✓ در عمل ماسک کردن ثبات B برای تعیین بیت هایی از A که قرار است پاک (صفر) شوند مورد استفاده قرار می گیرد.
- ✓ اگر ثبات A برابر با ۱۰۱۰ ۱۰۱۰ باشد و بخواهیم چهار بیت بزرگتر A را برابر ۱۰۰۱ کنیم :

0 1 1 0 1 0 1 0	A_t		0 0 0 0 1 0 1 0	A_t
0 0 0 0 1 1 1 1	B		1 0 0 1 0 0 0 0	B
				
0 0 0 0 1 0 1 0	A_{t+1}	$(A \leftarrow A \cdot B)$	1 0 0 1 1 0 1 0	A_{t+1} $(A \leftarrow A + B)$

- ✓ به ازای بیت هایی که در B مقدار صفر دارند، بیت های معادل آنها در A صفر می شود. بقیه بیت های A بدون تغییر می مانند.



عملیات درج

✓ برای وارد کردن رشته بیت مورد نظر به درون ثبات مورد استفاده قرار می‌گیرد.

✓ روش انجام عمل درج:

– ابتدا يك عمل ماسك براي پاك کردن بیت های مورد نظر انجام می شود.

– سپس يك عمل OR براي قرار دادن بیت های جدید مورد استفاده قرار می گیرد.

– مثال: فرض کنید می خواهیم ۱۰۱۰ را به قسمت کم ارزش ثبات A وارد کنیم.

1101 1000 1011 0001 مقدار اولیه ثبات A

1101 1000 1011 **1010** مقدار مورد انتظار ثبات A



عملیات درج



1101 1000 1011 0001

مقدار اولیه ثبات A

یرد.

1111 1111 1111 0000

ماسک

1101 1000 1011 0000

مقدار ثبات A پس از ماسک

0000 0000 0000 1010

بیت‌های درج در ثبات A

1101 1000 1011 1010

مقدار مورد انتظار ثبات A



عملیات پاک کردن

✓ در عمل پاک کردن اگر بیت های A و B مشابه بود، بیت معادل در A صفر می شود، در غیر این صورت بدون تغییر می ماند. اگر دو عدد برابر باشند نتیجه تمام صفر خواهد بود (برای مقایسه مناسب است)

1 1 0 0

A_t

1 1 0 0

B

0 0 0 0

A_{t+1}

$(A \leftarrow A \oplus B)$

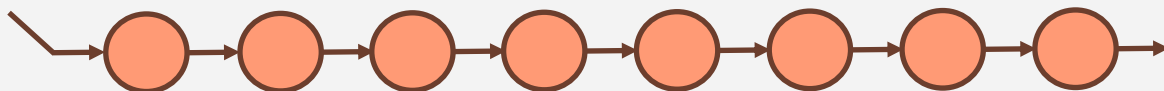


ریز عمل شیفت در سیستم‌های دیجیتال

- ۱ - شیفت منطقی
- ۲ - شیفت چرخشی
- ۳ - شیفت حسابی

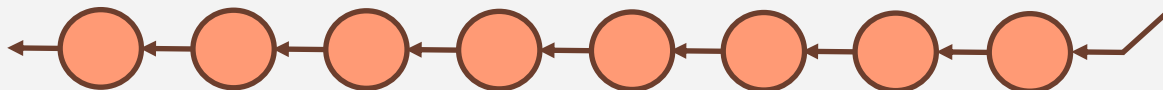
✓ تفاوت انواع شیفت نامبرده در بیت ورودی سریال است.

ورودی سریال



شیفت به راست

ورودی سریال



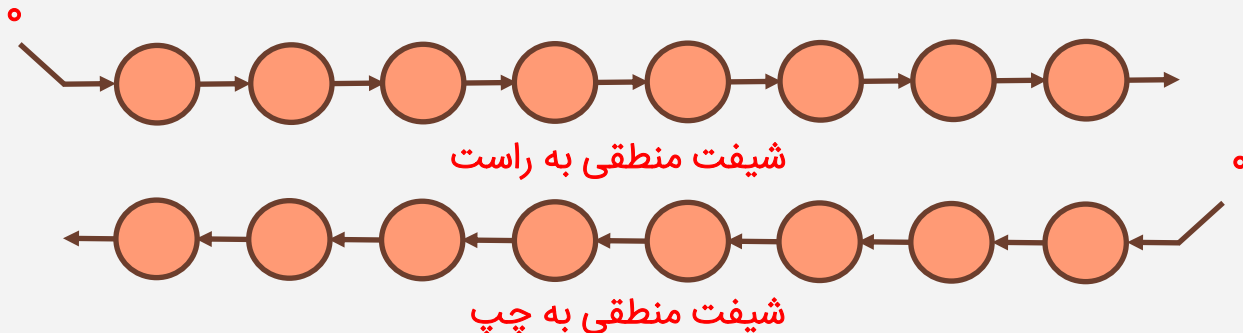
شیفت به چپ



شیفت منطقی

✓ شیفت منطقی برای اعداد بی علامت مورد استفاده قرار می‌گیرد.

✓ بیت ورودی، در این نوع شیفت صفر است.



✓ در زبان انتقال ثبات (RTL) از علائم زیر استفاده می‌شود:

- shl شیفت منطقی به چپ

مثال

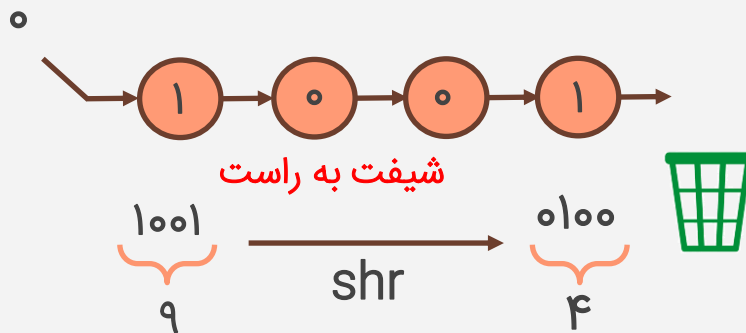
$R_2 \leftarrow \text{shl } R_2$

- shr شیفت منطقی به راست

$R_3 \leftarrow \text{shr } R_3$



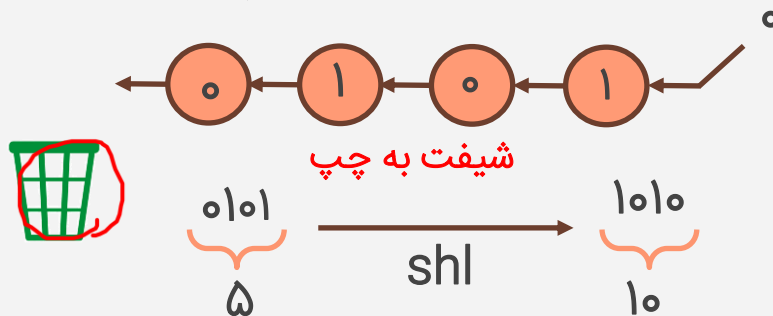
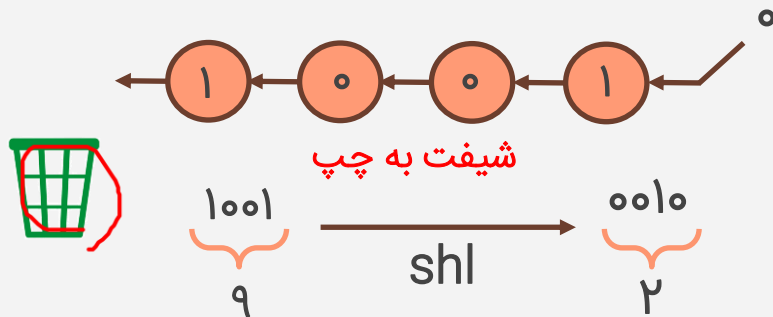
شیفت منطقی به راست



✓ شیفت منطقی به راست، معادل جزء صحیح تقسیم بر ۲ است.



شیفت منطقی به چپ

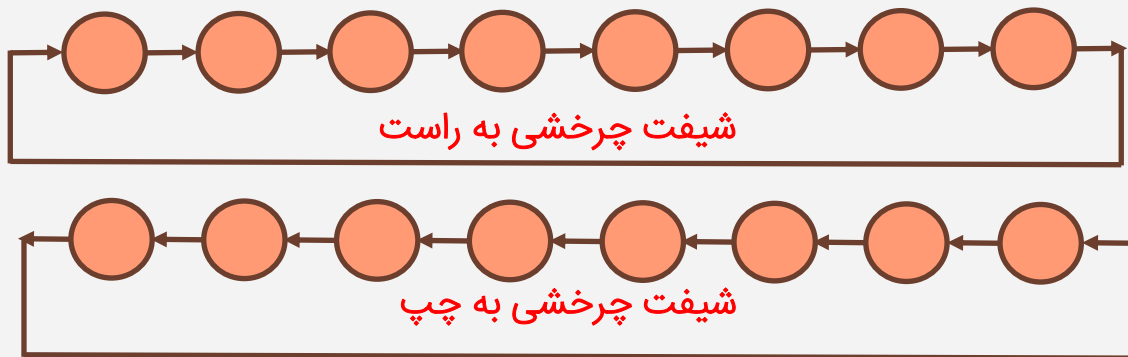


✓ شیفت منطقی به چپ، معادل ضرب در ۲ است (به شرطی که سرریز رخ ندهد).



شیفت چرخشی

✓ در شیفت چرخشی بیت ورودی سریال، بیت خروجی از سمت دیگر ثابت است.



✓ در زبان انتقال ثبات از علائم زیر استفاده می شود

- شیفت چرخشی به چپ cil

مثال

$$R^2 \leftarrow \text{cil } R^2$$

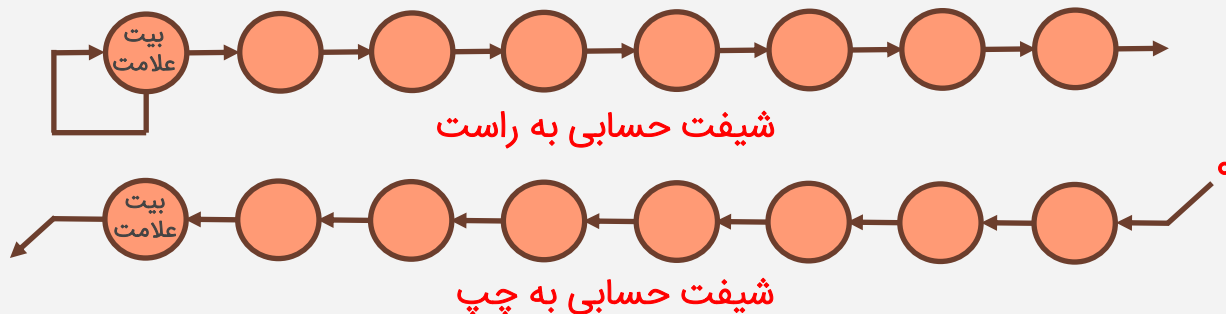
- شیفت چرخشی به راست cir

$$R^3 \leftarrow \text{cir } R^3$$



شیفت حسابی

- ✓ شیفت حسابی برای اعداد علامت دار مکمل ۲ مناسب است.
- ✓ مهمترین ویژگی شیفت حسابی آن است که به هنگام شیفت (ضرب و تقسیم) علامت ثبات را حفظ می کند.



- ✓ در زبان انتقال ثبات از علائم زیر استفاده می شود

- شیفت چرخشی به چپ ashl

- شیفت چرخشی به راست ashr

مثال

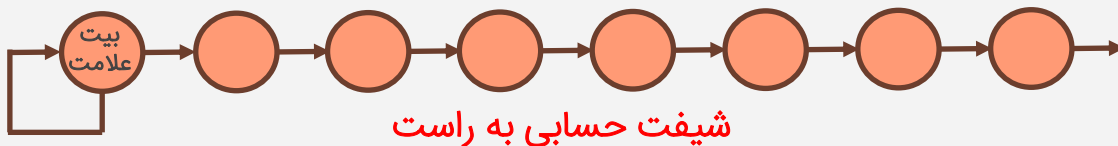
$R_2 \leftarrow \text{ashl } R_2$

$R_3 \leftarrow \text{ashr } R_3$

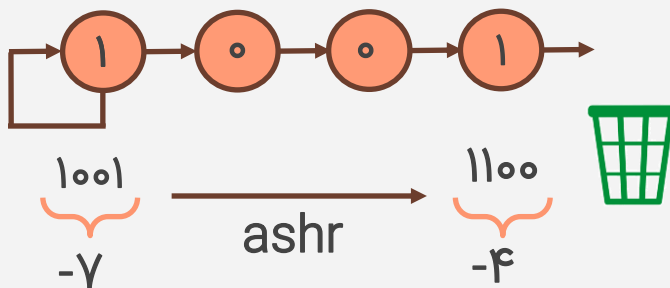


شیفت حسابی به راست

در شیفت حسابی به راست، علامت عدد از سمت چپ وارد می شود

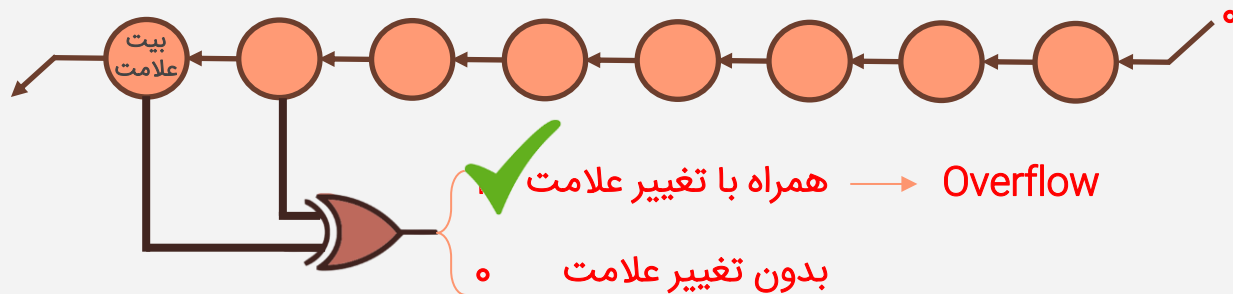


شیفت حسابی به راست معادل جزء صحیح تقسیم عدد درون ثبات بر ۲ است

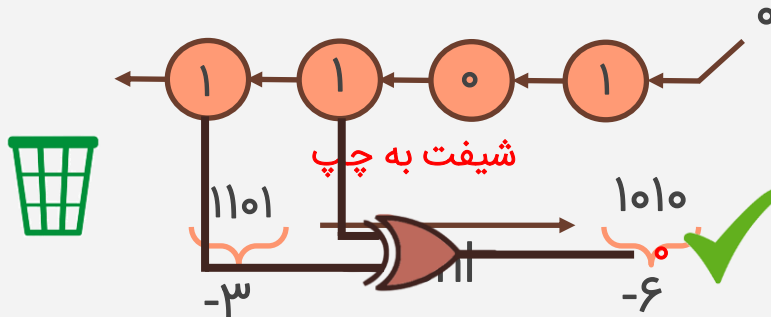




شیفت حسابی به چپ

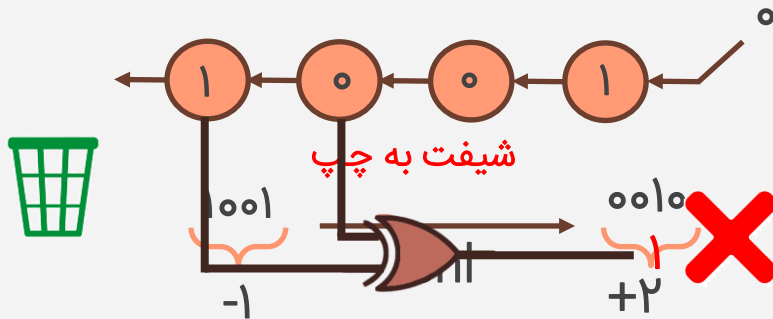


- ✓ در شیفت حسابی به چپ نیز مثل شیفت منطقی به چپ از سمت راست صفر وارد می شود.
- ✓ شیفت حسابی به چپ عدد درون ثبات را در ۲ ضرب می کند.



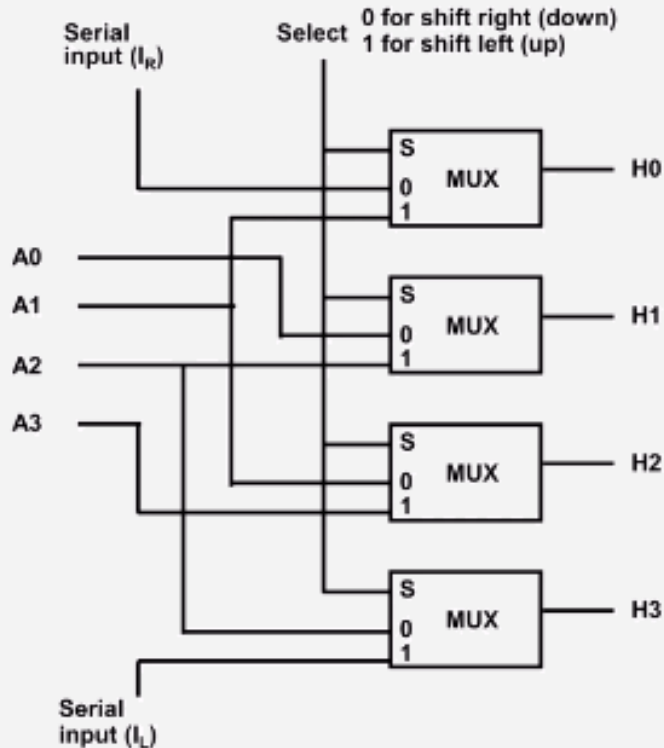


شیفت حسابی به چپ



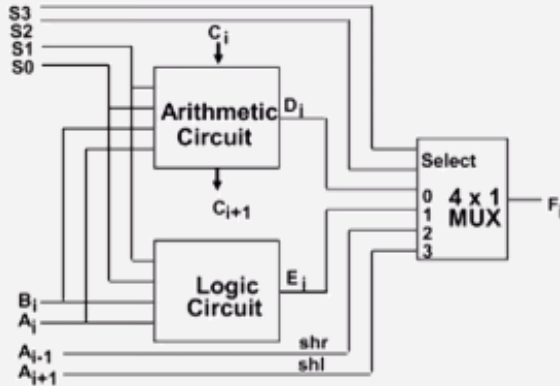


پیاده‌سازی سخت افزاری ریز عمل‌های شیفت





واحد عمليات شيفت، منطقي، حسابي



S3	S2	S1	S0	Cin	Operation	Function
0	0	0	0	0	$F = A$	Transfer A
0	0	0	0	1	$F = A + 1$	Increment A
0	0	0	1	0	$F = A + B$	Addition
0	0	0	1	1	$F = A + B + 1$	Add with carry
0	0	1	0	0	$F = A + B'$	Subtract with borrow
0	0	1	0	1	$F = A + B' + 1$	Subtraction
0	0	1	1	0	$F = A - 1$	Decrement A
0	0	1	1	1	$F = A$	Transfer A
0	1	0	0	X	$F = A \wedge B$	AND
0	1	0	1	X	$F = A \vee B$	OR
0	1	1	0	X	$F = A \oplus B$	XOR
0	1	1	1	X	$F = A'$	Complement A
1	0	X	X	X	$F = \text{shr } A$	Shift right A into F
1	1	X	X	X	$F = \text{shl } A$	Shift left A into F



razeghizade@gmail.com

Razeghizade.pudica.ir

CREADITS: This presentation was created by M.Razeghizadeh
Please keep this slide for attribution