ریزپردازنده و زبان اسمبلی

ارائه دهنده:

هادی اولیا

رئوس مطالب

- مقدمه ای بر معماری کامپیوتر و معرفی میکروپروسسور ها
- برنامه نویسی به زبان اسمبلی
- برنامه نویسی میکروکنترلر AVR

عناوین

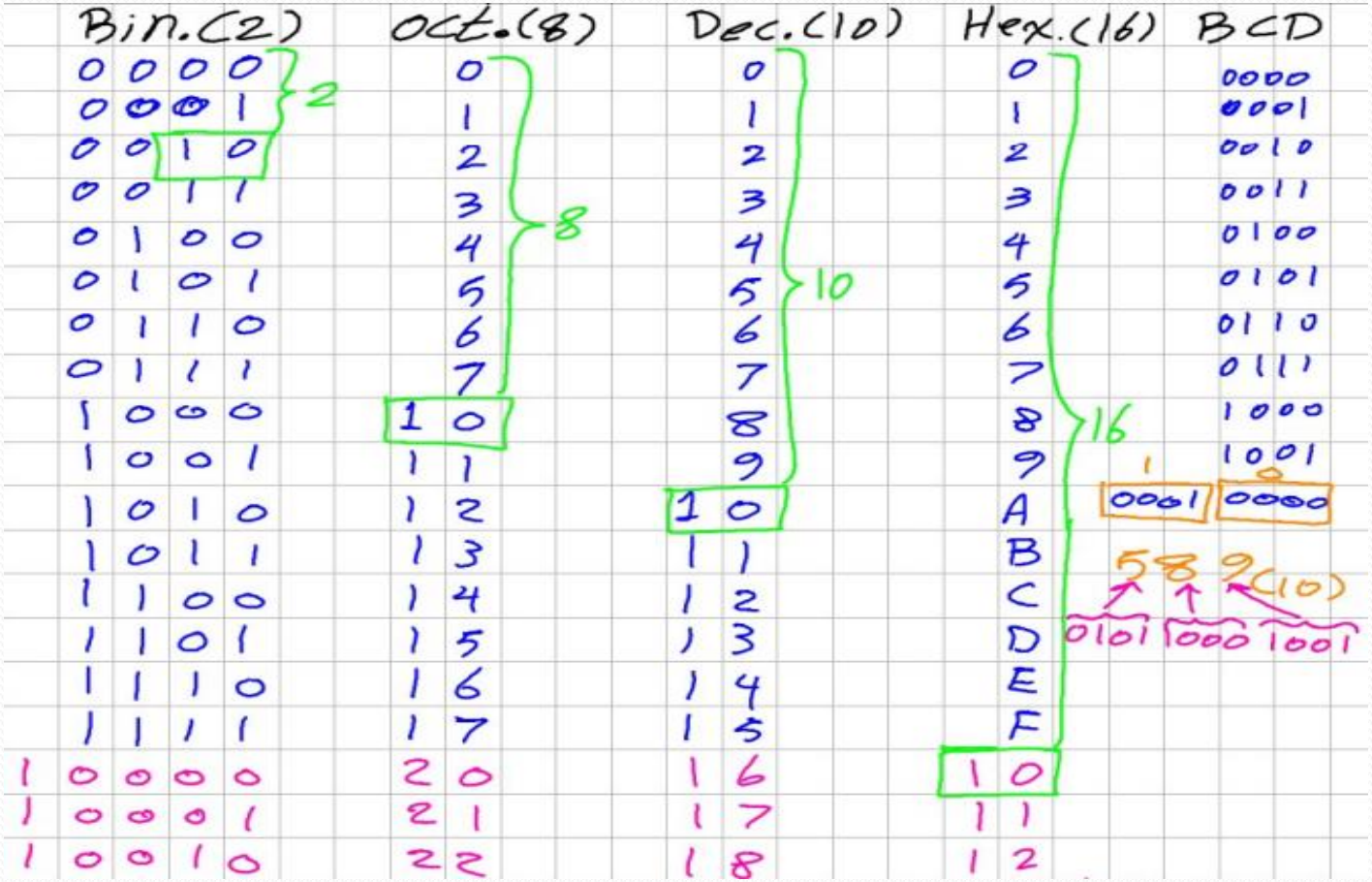
- معماری کامپیوتر
- سیستم اعداد
- ساختار کامپیوتر
- انواع CPU

مراجع (۱)

- معماری کامپیوتر، تالیف: موریس مانو، ترجمه: دکتر سید رضی
- مدار های واسط، تالیف: مزیدی، ترجمه: دکتر سپید نام
- زبان اسمبلی، تالیف: مزیدی، ترجمه: دکتر سپید نام

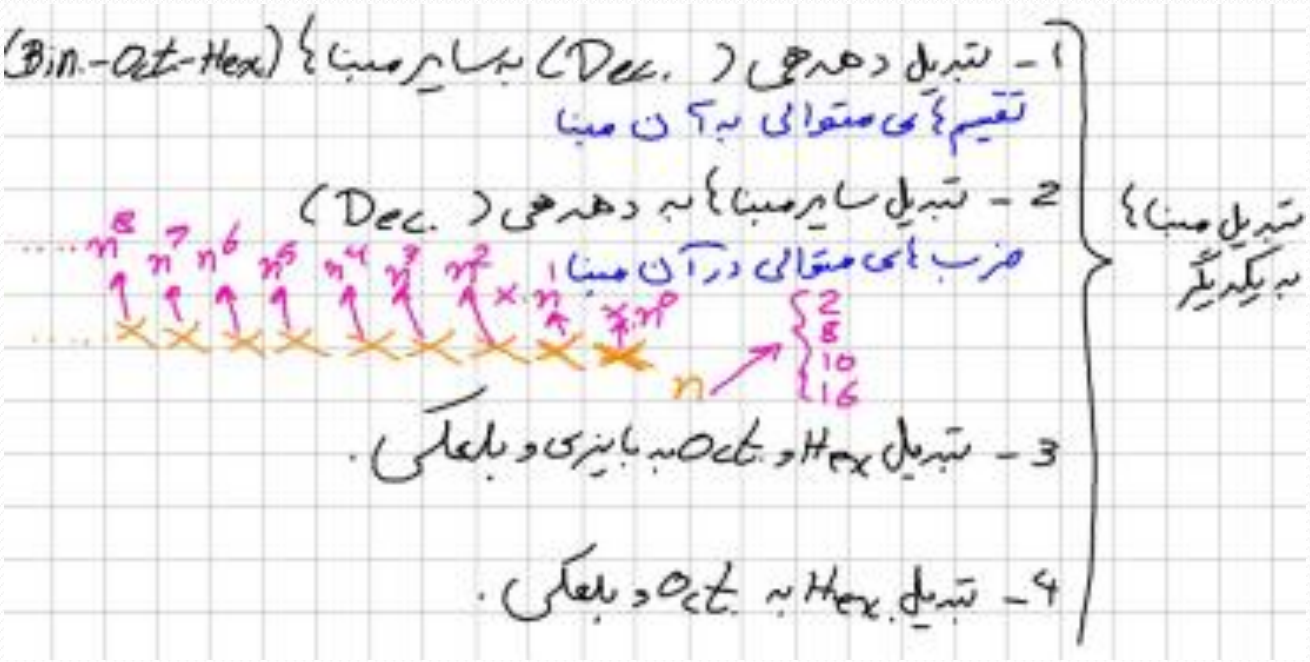
سیستم اعداد

نمایش اعداد



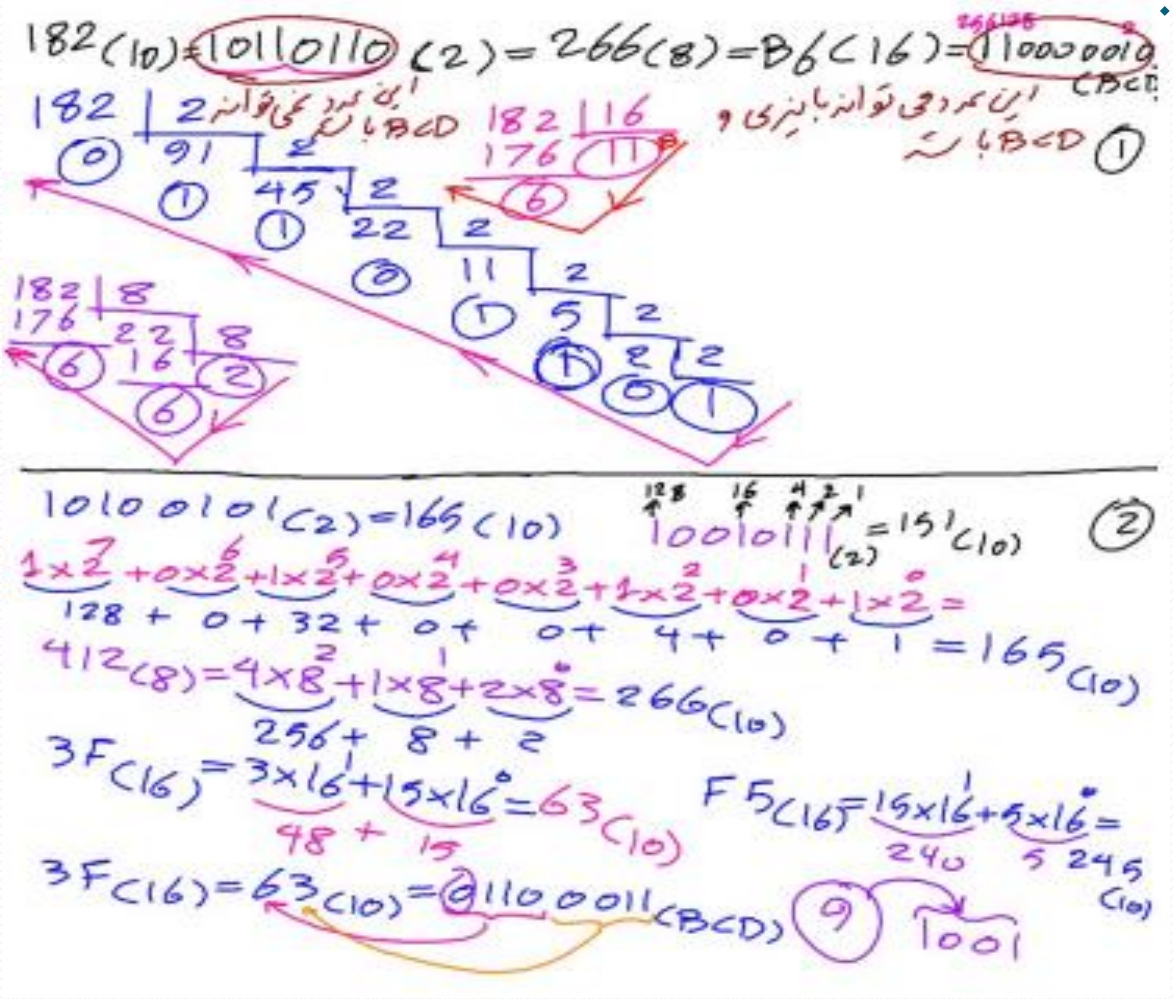
سیستم اعداد

تبدیل مبنا

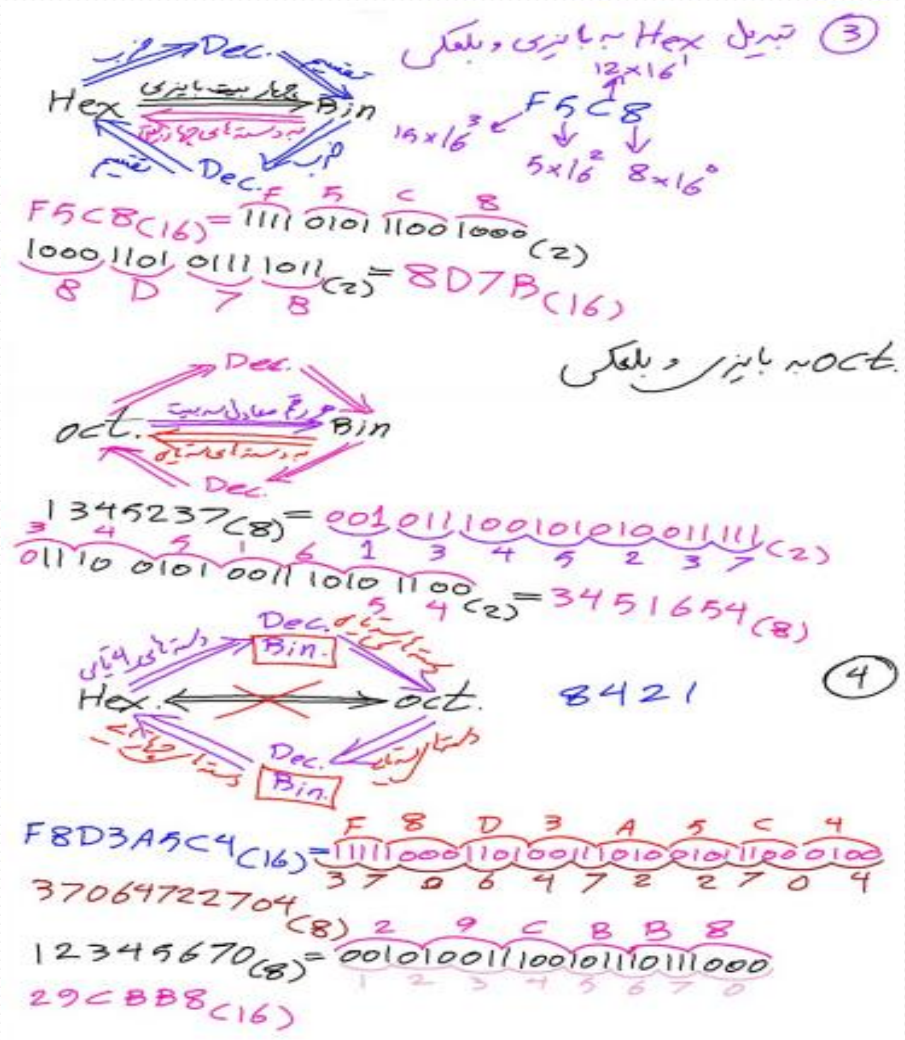


سیستم اعداد

تبدیل مبنا

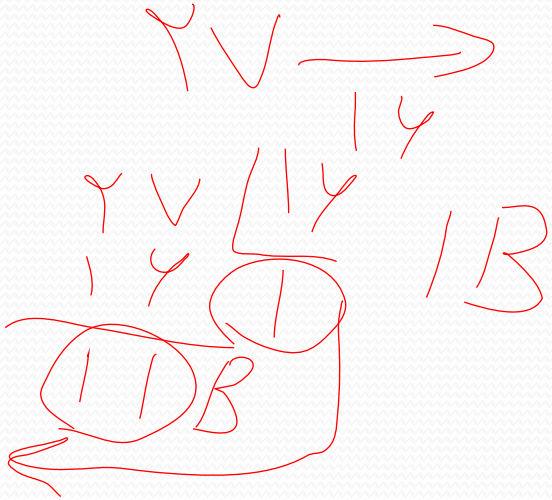


سیستم اعداد
تبدیل مبنا



سیستم اعداد

عملیات جمع و تفریق



4E85 H	4E85 H
+ 2D07 H	- 2D07 H
7B8C H	217E H

نمایش اعداد علامت دار

- یک عدد علامت دار دارای:
  - ✓ اندازه (قدر مطلق مقدار)
  - ✓ علامت (مثبت یا منفی)
  - ✓ هر مقداری باید باینری باشد یعنی دنباله ای از صفر ها و یک ها

- دو روش اعداد علامت دار
  - ❖ نمایش علامت و اندازه
  - ❖ نمایش مکمل

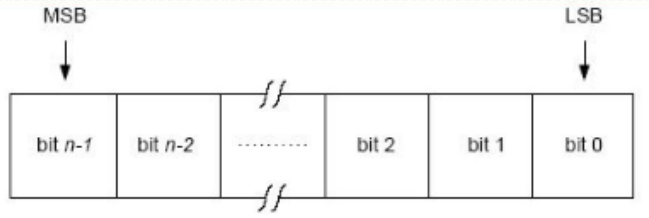
نمایش اعداد علامت دار

نمایش علامت و اندازه

با ارزشترین بیت (MSB) بیت علامت در نظر گرفته می شود.

‘0’ --> عدد مثبت

‘1’ --> عدد منفی



مابقی بیت ها اندازه را نمایش می دهند.

کل تعداد اعداد قابل نمایش با یک رجیستر **n** بیتی می توان نشان داد برابر است با

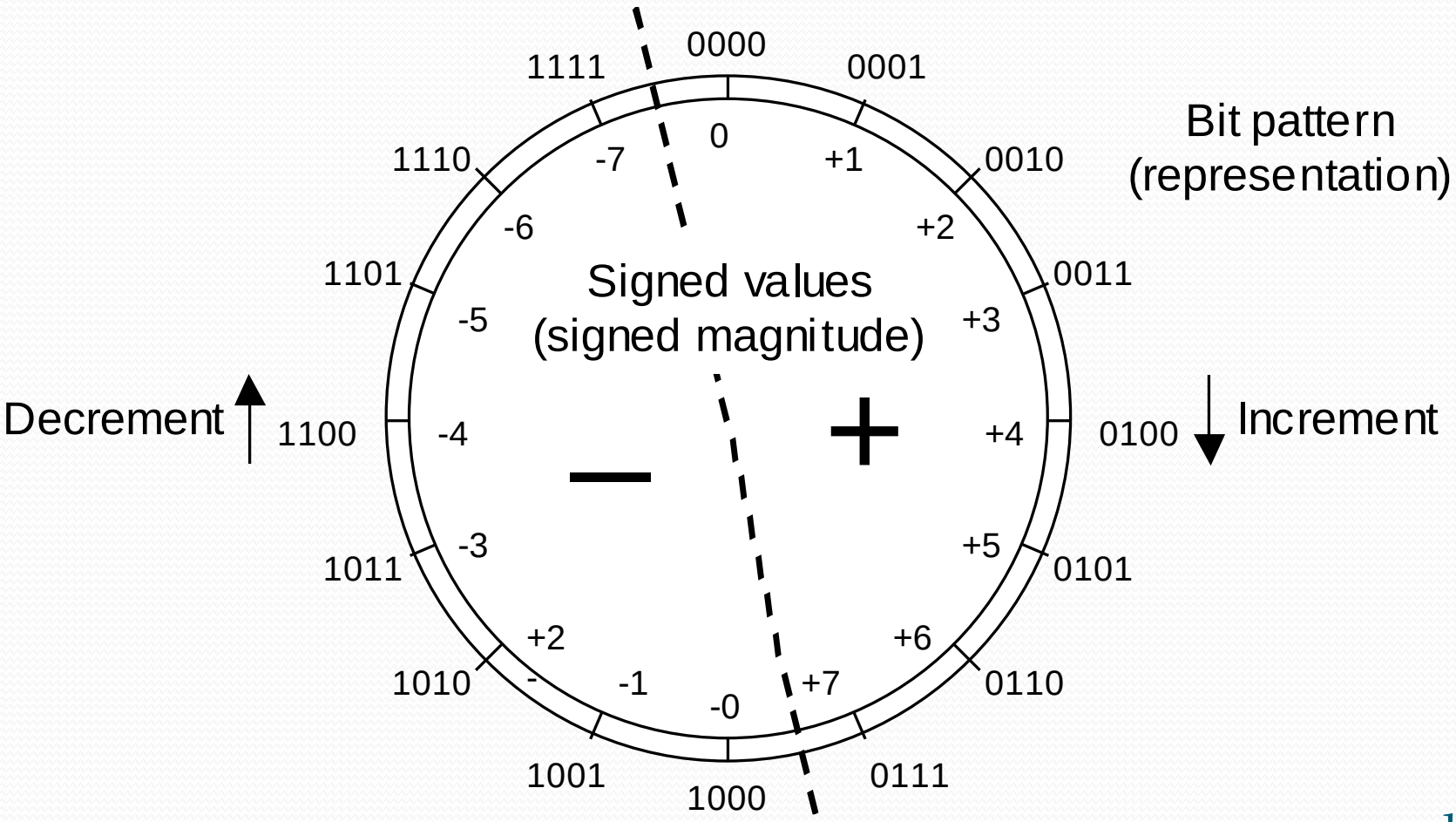
2^n-1

رنج اعداد:

-(2^n-1-1), … , -1, -0, +0, +1, …, +(2^n-1-1)

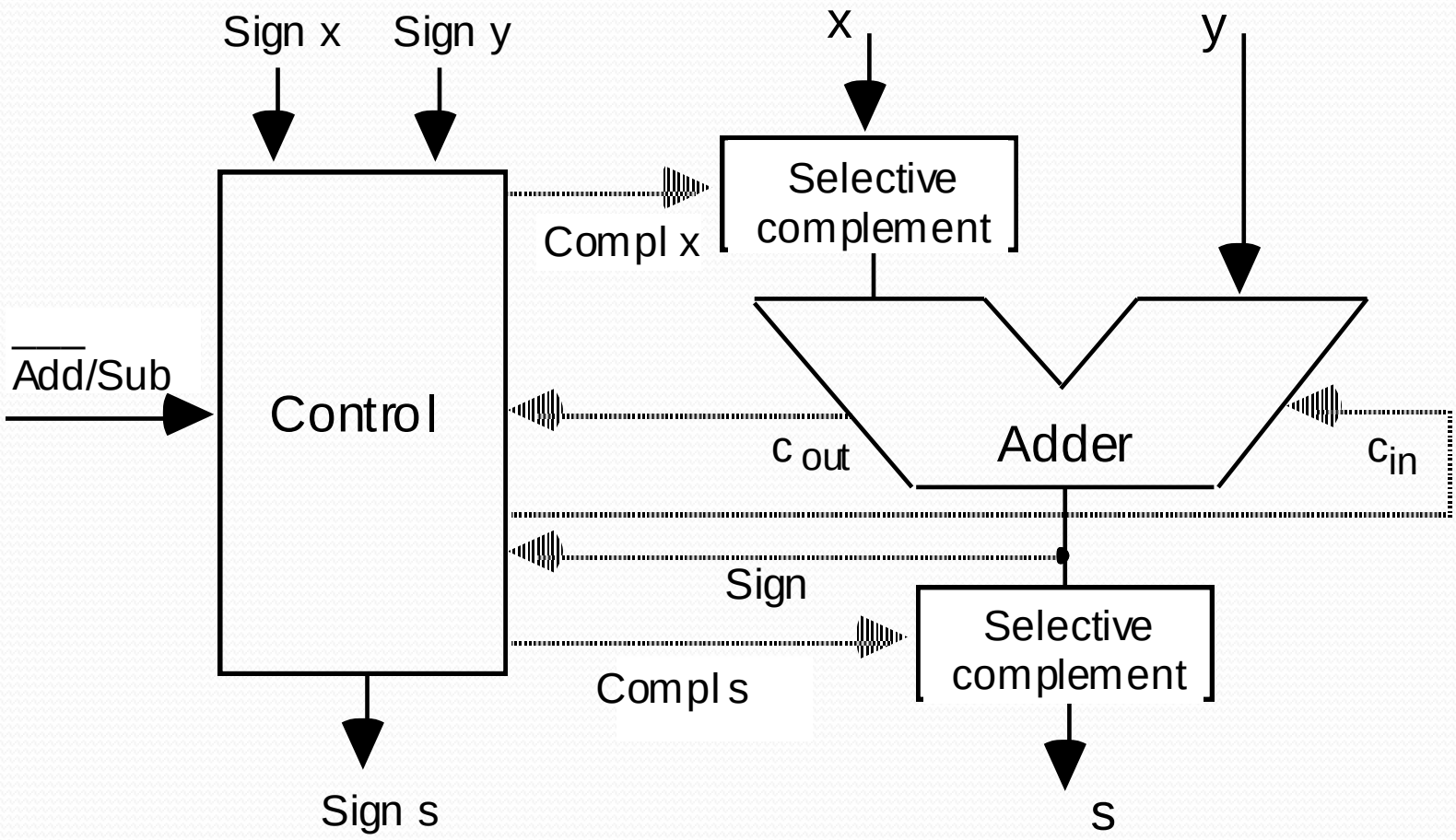
نمایش اعداد علامت دار

نمایش علامت و اندازه برای اعداد ۴ بیتی



نمایش علامت و اندازه

جمع کننده



نمایش اعداد علامت دار

نمایش مکمل

دو نوع مکمل برای مبنای ۲

- مکمل ۱: تمام بیت ها معکوس شوند.

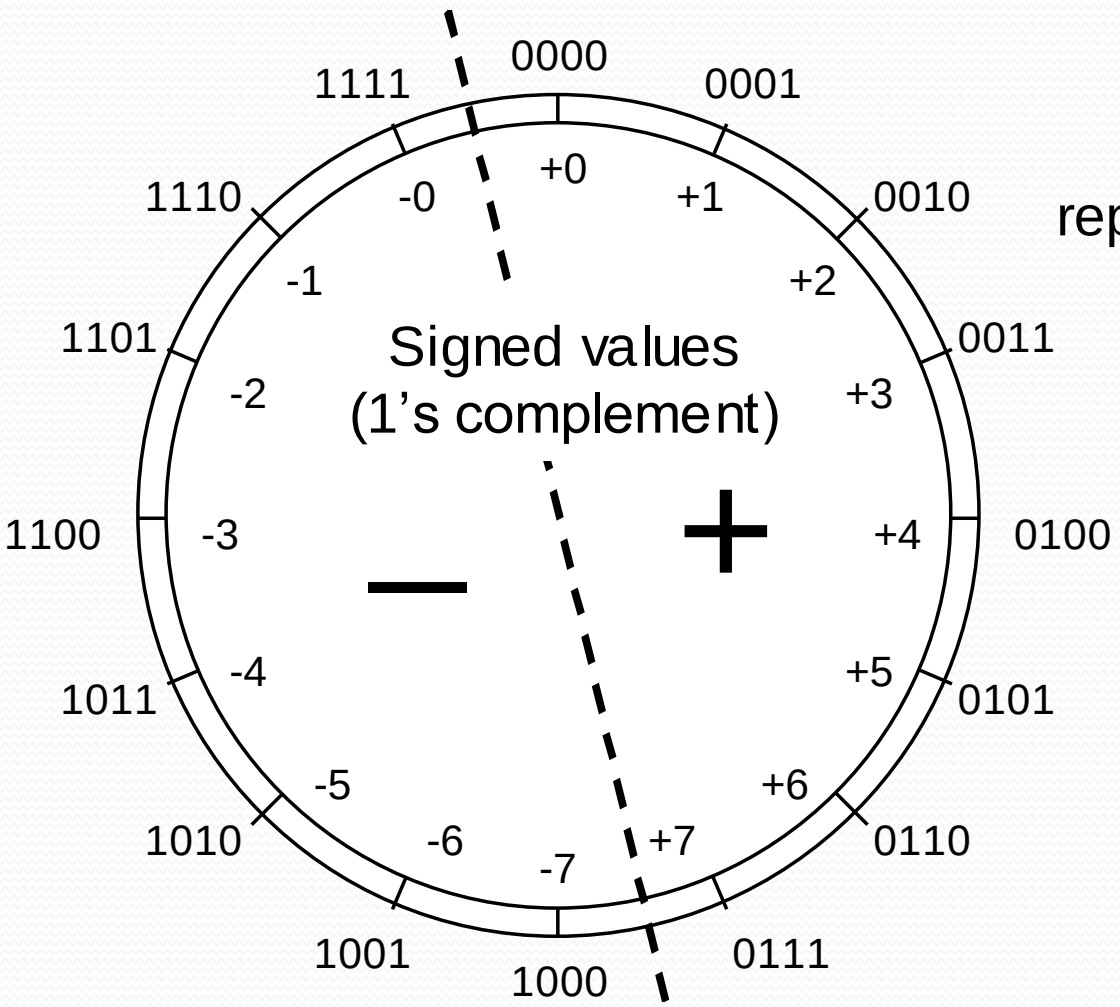
- مکمل ۲: به دو روش انجام می شود

✓ یک واحد به مکمل یک اضافه شود.

✓ شروع از سمت راست و یافتن و نگه داشتن اولین یک، سپس تمامی ارقام سمت چپ آن معکوس می شوند.

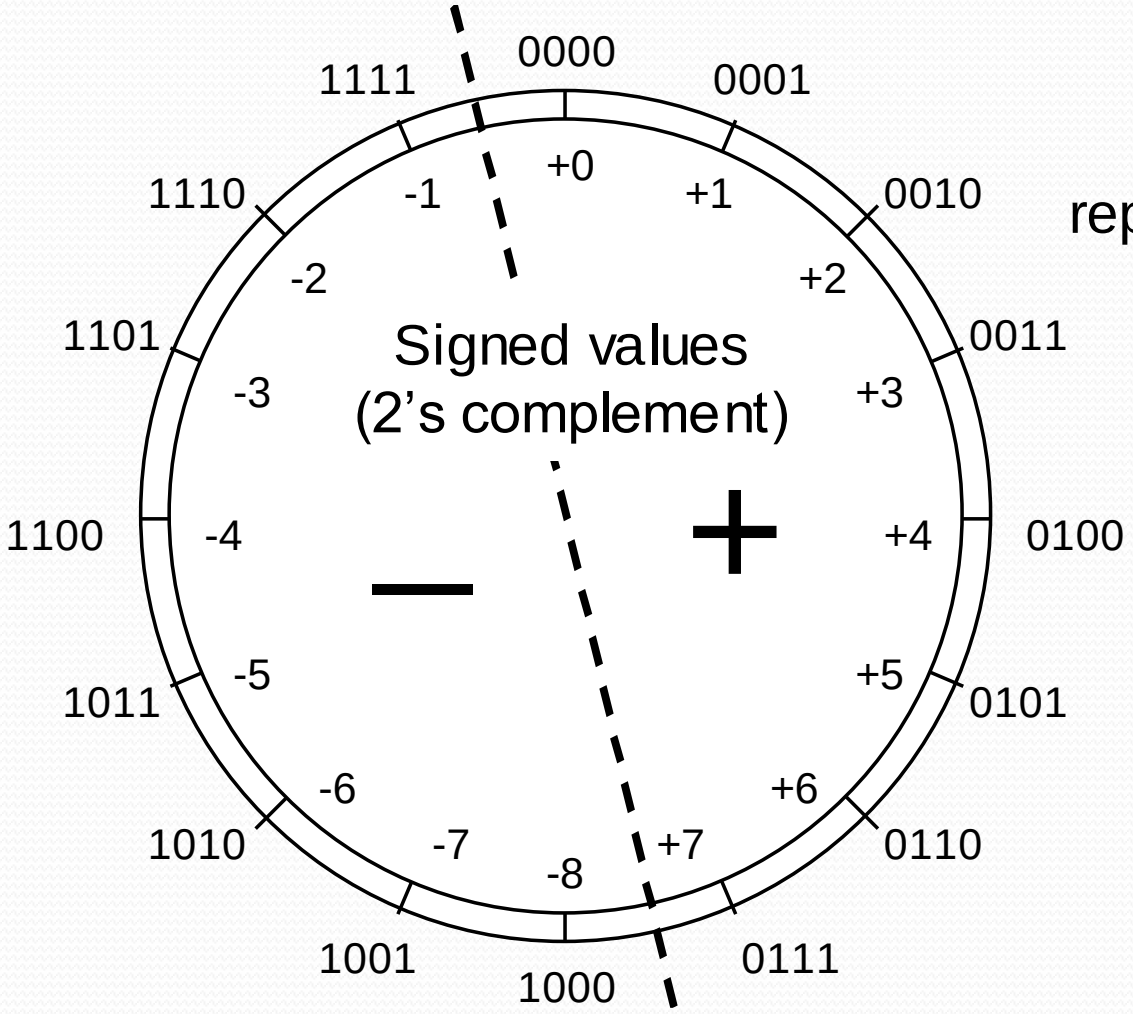
نمایش مکمل ۱

اعداد ۴ بیتی



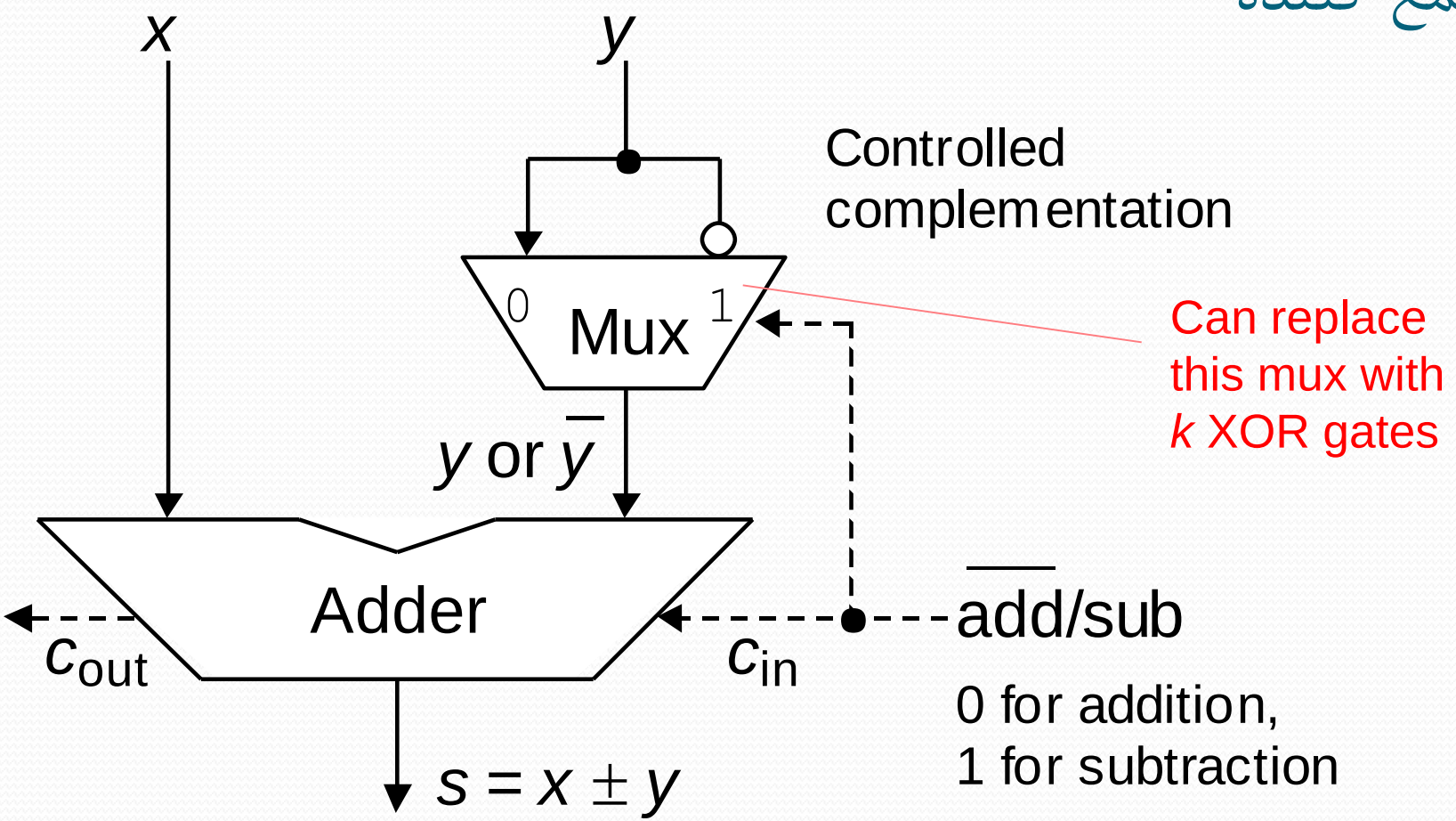
نمایش مکمل ۲

اعداد ۴ بیتی

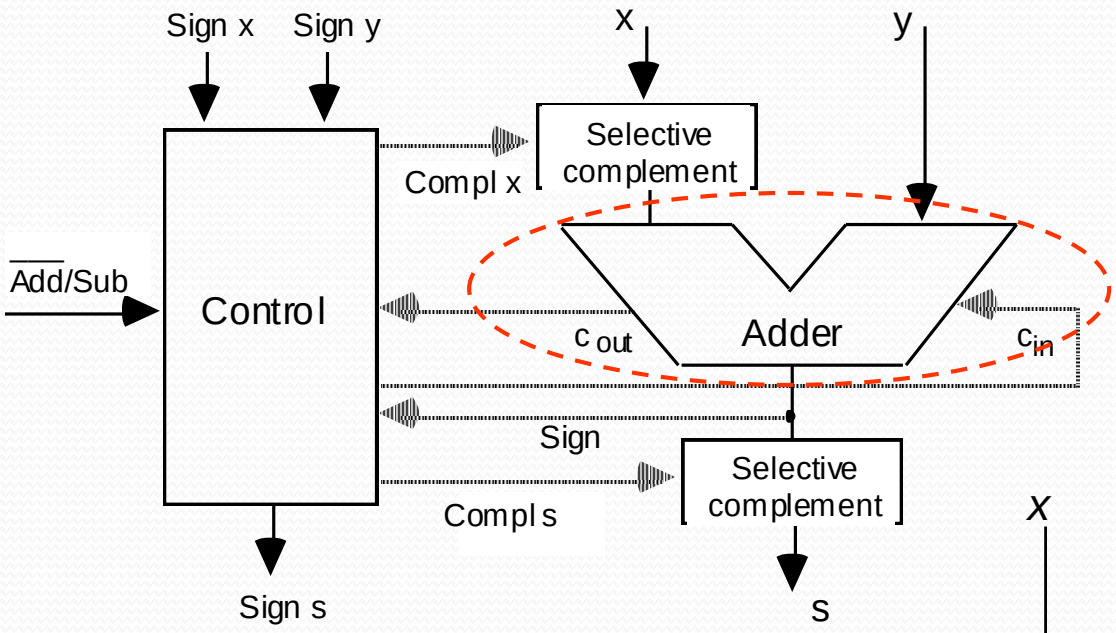


Unsigned representations

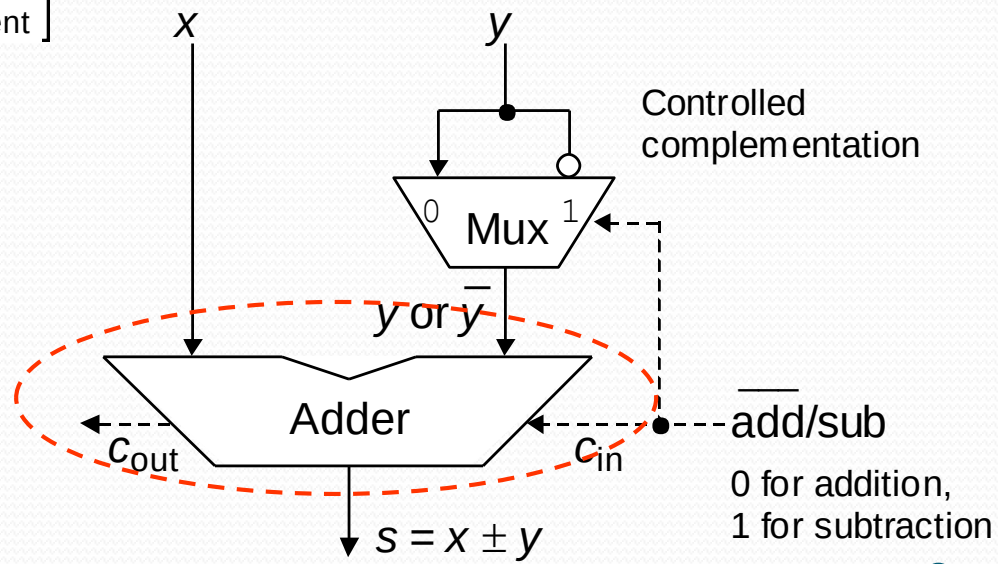
نمایش مکمل ۲
جمع کننده



مقایسه عمل جمع در سیستم علامت و مقدار با سیستم مکمل ۲



Signed-magnitude adder/subtractor is significantly more complex than a simple adder



2's-complement adder/subtractor needs very little hardware other than a simple adder

نمایش اعداد علامت دار

خلاصه

- ✓ اعداد مثبت در تمامی نمایش ها یکسان هستند بعبارتی بیت MSB آنها صفر است.
- ✓ تنها نمایش اعداد با مکمل ۲ یک صفر دارد.
- ✓ مکمل ۲ قادر است تا یک عدد منفی بیشتر را نمایش دهد.
- ✓ عملیات جمع/تفریق در مکمل ۲ بصورت ساده تری قابل پیاده سازی است.

کد اسکی (ASCII)

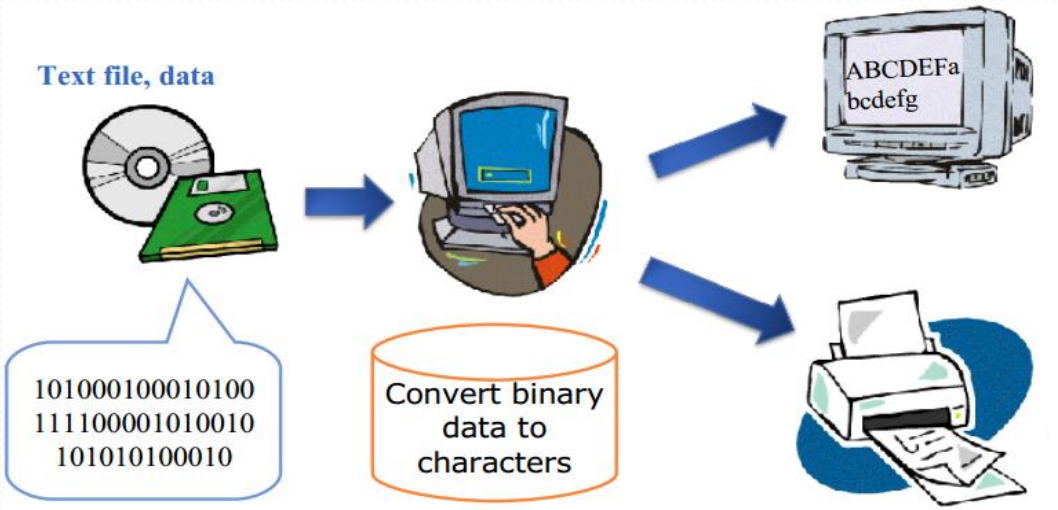
- اختصار ASCII

American Standard Code for Information Interchange (ASCII)

- یک کد استاندارد صنعتی که توسط موسسه استاندارد های ملی امریکا (ANSI) در سال ۱۹۶۳ ارائه شده است.
- ASCII یک کد هفت بیتی (از ۰ تا ۱۲۷) است که یک مقدار باینری را به حروف، اعداد و کاراکتر ها اختصاص می دهد.
- کد های ASCII جهت نمایش متون و کاراکتر های کنترلی در کامپیوتر ها، تجهیزات ارتباطی و افزاره های دیگر که با متن کار می کنند، استفاده می شود.
- اغلب کامپیوتر ها از کد ASCII جهت نمایش متون در انتقال داده از یک کامپیوتر به کامپیوتر دیگر استفاده می کنند.

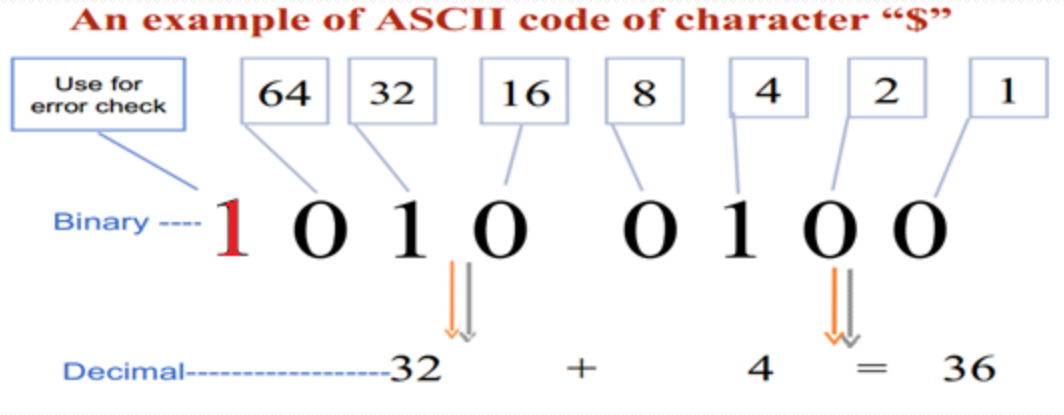
کد ASCII

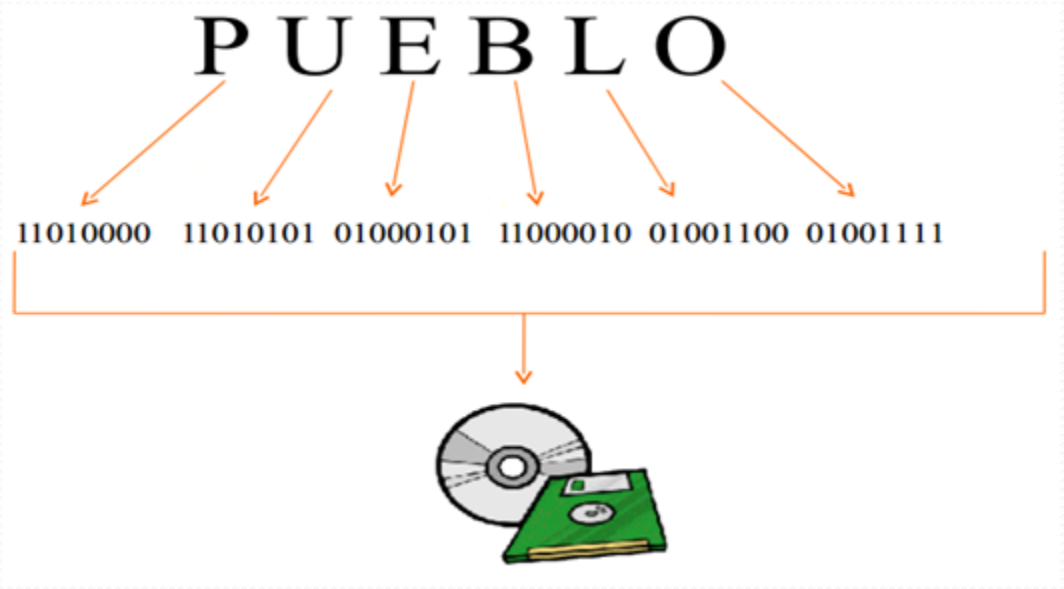
- کامپیوتر ها از یک جدول ASCII به منظور تبدیل داده باینری به حروف یا کاراکتر ها استفاده می کنند. از اینرو ما می توانیم آنها را بصورت بصری ببینیم.



Character	Binary code	Character	Binary code
A	100 0001	0	011 0000
B	100 0010	1	011 0001
C	100 0011	2	011 0010
D	100 0100	3	011 0011
E	100 0101	4	011 0100
F	100 0110	5	011 0101
G	100 0111	6	011 0110
H	100 1000	7	011 0111
I	100 1001	8	011 1000
J	100 1010	9	011 1001
K	100 1011		
L	100 1100		
M	100 1101	space	010 0000
N	100 1110	.	010 1110
O	100 1111	(	010 1000
P	101 0000	+	010 1011
Q	101 0001	$	010 0100
R	101 0010	*	010 1010
S	101 0011	)	010 1001
T	101 0100	-	010 1101
U	101 0101	/	010 1111
V	101 0110	,	010 1100
W	101 0111	=	011 1101
X	101 1000		
Y	101 1001		
Z	101 1010		

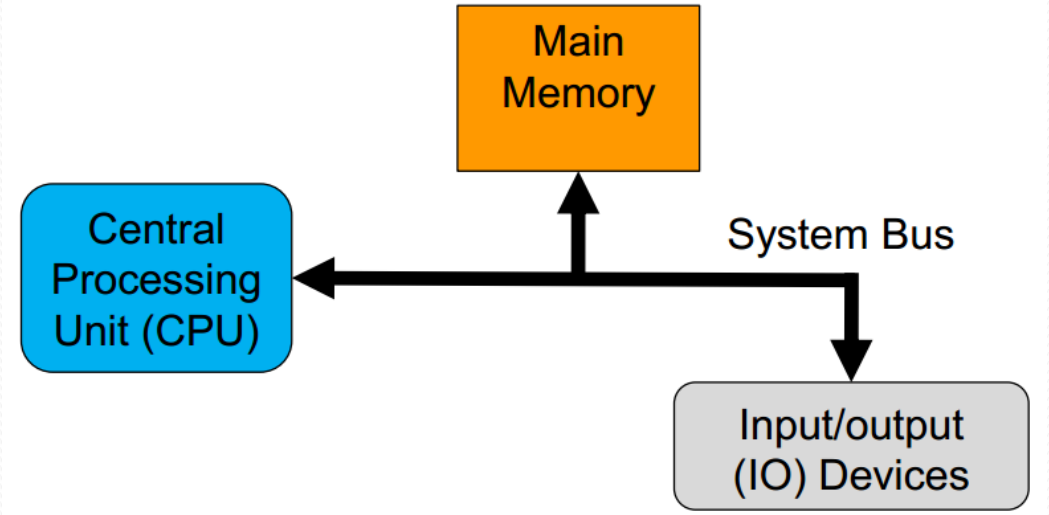
کد ASCII: مثال





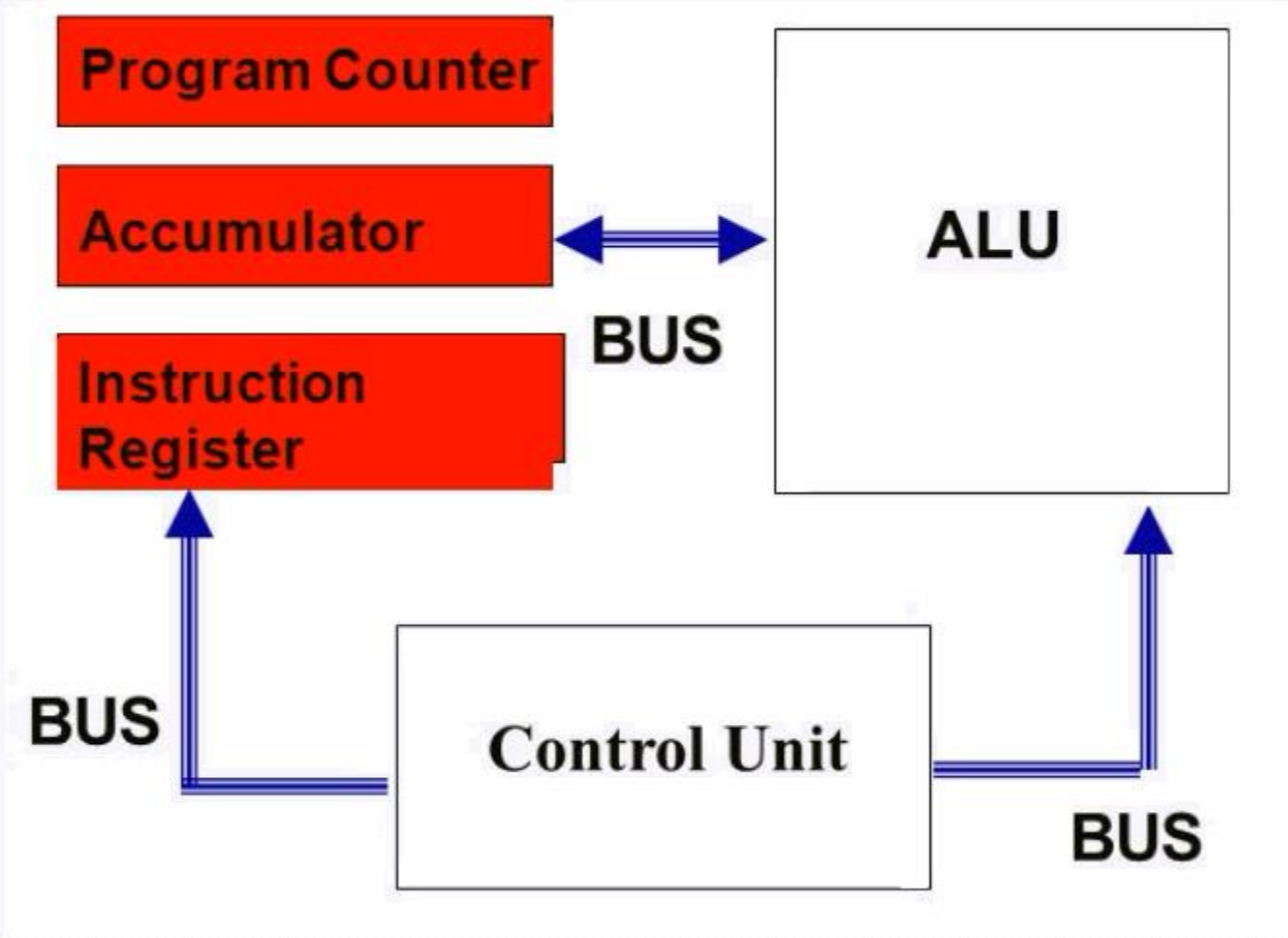
Character	Binary code	Character	Binary code
A	100 0001	0	011 0000
B	100 0010	1	011 0001
C	100 0011	2	011 0010
D	100 0100	3	011 0011
E	100 0101	4	011 0100
F	100 0110	5	011 0101
G	100 0111	6	011 0110
H	100 1000	7	011 0111
I	100 1001	8	011 1000
J	100 1010	9	011 1001
K	100 1011		
L	100 1100		
M	100 1101	space	010 0000
N	100 1110	.	010 1110
O	100 1111	(	010 1000
P	101 0000	+	010 1011
Q	101 0001	$	010 0100
R	101 0010	*	010 1010
S	101 0011	)	010 1001
T	101 0100	–	010 1101
U	101 0101	/	010 1111
V	101 0110	,	010 1100
W	101 0111	=	011 1101
X	101 1000		
Y	101 1001		
Z	101 1010		

ساختار کامپیوتر
نگاه سطح بالا به یک کامپیوتر

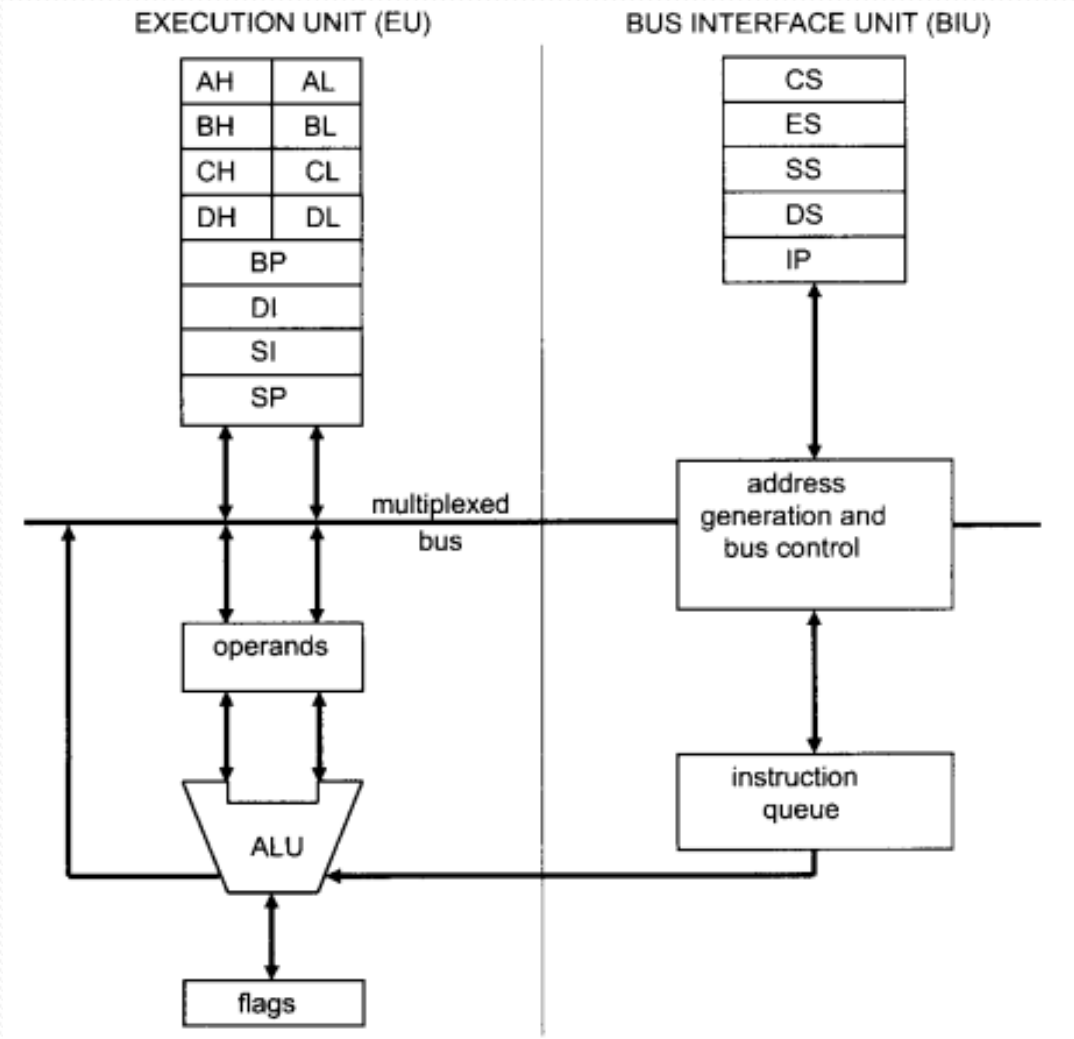


- CPU: اجرای دستورالعمل ها
- حافظه: ذخیره برنامه و داده
- افزاره های IO: دریافت ورودی ها و تولید خروجی ها
- باس (Bus): اتصال واحد ها با انتقال داده

واحد پردازنده مرکزی (CPU)



8086/88 CPU


EXECUTION UNIT (EU)
BUS INTERFACE UNIT (BIU)
AH
AL
BH
BL
CH
CL
DH
DL
BP
DI
SI
SP
CS
ES
SS
DS
IP
multiplexed
bus
address
generation and
bus control
operands
instruction
queue
ALU
flags

واحد پردازنده مرکزی (CPU)
اجزای CPU

۱) ثبات (Register): مکانی در داخل CPU که اطلاعات بصورت موقت در آن نوشته و یا از آن خوانده می شود.

- ثبات های عمومی: A، B، C، D
- ثبات های خصوصی: IR، IP (PC)، SP

۲) واحد محاسبه و منطق (ALU): بخشی از سیستم که در آن کلیه محاسبات عددی و منطقی یک پردازنده انجام می شود.

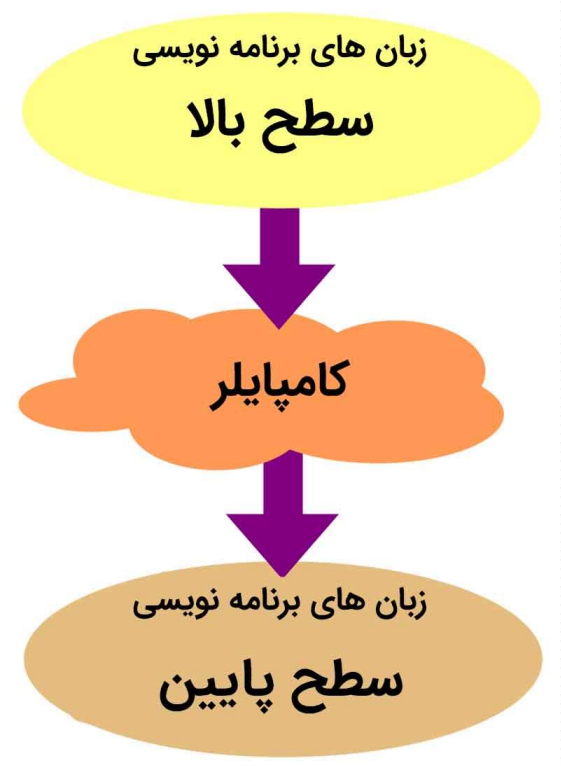
واحد پردازنده مرکزی (CPU)
اجزای CPU

۳) باس یا گذرگاه (Bus): تعدادی خطوط انتقال که از طریق آن آدرس، داده و فرمان های کنترلی منتقل می گردد.

- خطوط آدرس
- خطوط داده
- خطوط کنترل

۴) واحد کنترل (control unit): بخشی از CPU که دستورات پردازنده را هدایت می کند. این واحد طریقه پاسخگویی به دستورات برنامه را توسط حافظه، ALU و افزاره های ورودی/خروجی تعیین می نماید.

زبان های برنامه نویسی



زبان سطح بالا — A+B

زبان اسمبلی — ADD A,B

زبان ماشین — 100011100001010

واحد پردازنده مرکزی (CPU)
مراحل برقراری یک دستورالعمل

۱) واکشی (Fetch): به آدرسی که IP به آن اشاره می کند مراجعه و کد دستورالعمل را می آورد.

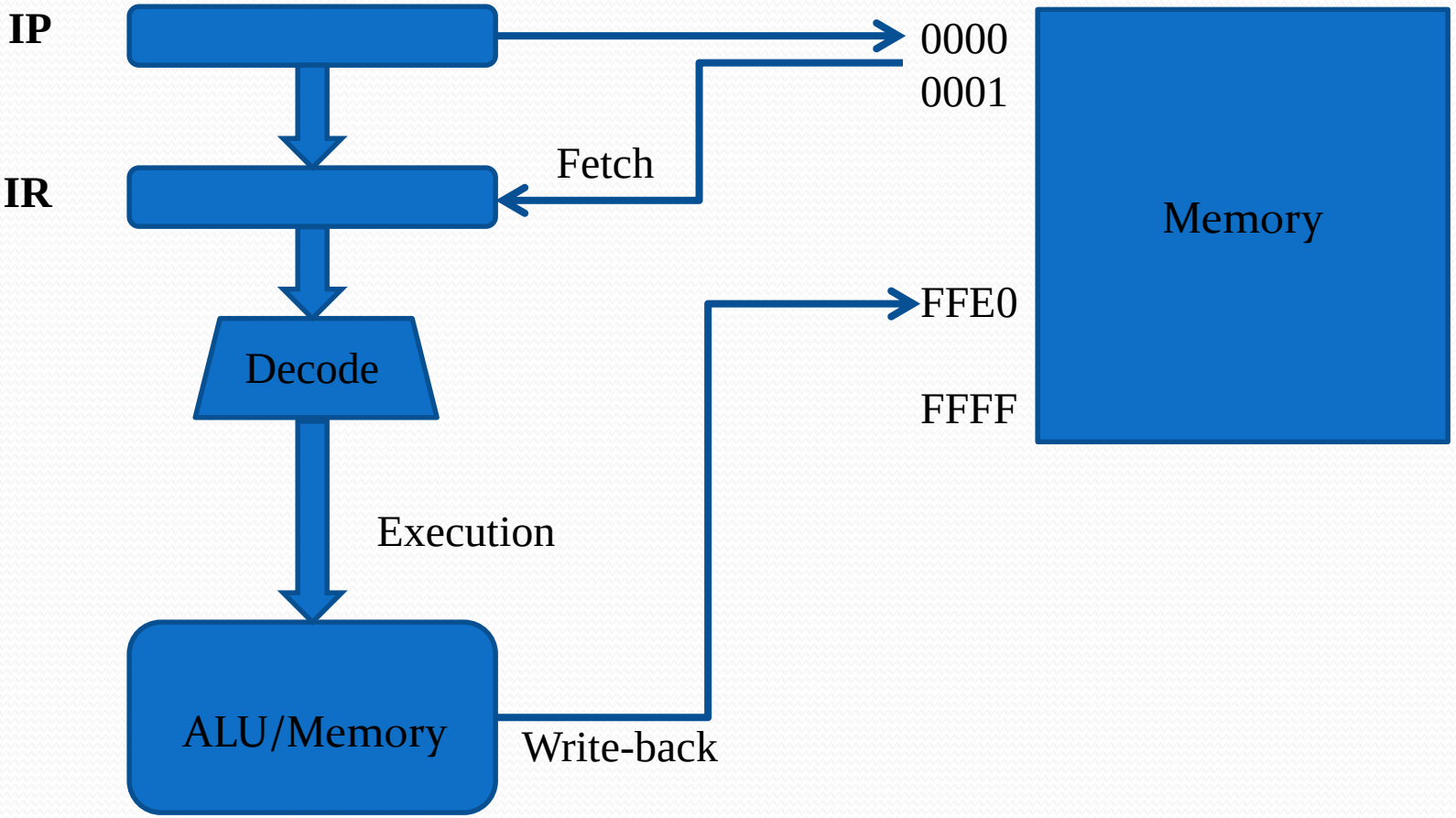
۲) رمزگشایی (Decode): یعنی تفسیر دستور بعبارتی چه عملی باید اجرا شود.

۳) اجرا (Execution): دستورالعمل ها اجرا می شوند.

۴) نوشتن (Write-back): نوشتن حاصل عملیات

واحد پردازنده مرکزی (CPU)

مراحل برقراری یک دستورالعمل



معماری مجموعه دستورات

- معادل لاتین

- Instruction set architecture (ISA)
- هدف طراحی مجموعه دستورات برای ریز پردازنده است.
- افزایش تعداد دستورات تبدیل زبان سطح بالا به پایین را تسهیل می نماید.
- انواع:

✓ CISC (Complex instruction set architecture)

✓ RISC (Reduced instruction set architecture)

معماری CISC

- هدف: تهیه یک دستور العمل برای هر عبارتی که در سطح بالا نوشته می شود.
- مثال ها: ۸۰۸۶، ۸۰۵۱
- ویژگی های معماری CISC

✓ تعداد زیاد دستور العمل ها (معمولا بین ۱۰۰ تا ۲۵۰ دستور)

✓ دستور العمل هایی که کار های خاصی را انجام می دهند ولی بندرت بکار می روند.

✓ انواع مختلفی از روش های آدرس دهی موجود است (معمولا بین ۵ تا ۲۰ روش مختلف)

✓ قالب دستورات با طول متغیر است.

✓ دستوراتی که عملوند ها را در حافظه دستکاری می کنند.

معماری RISC

- هدف: ارجاع سریع تر CPU و کاهش ارجاعات به حافظه
- مثال ها: تکنولوژی های مطرح امروزی
- ویژگی های معماری RISC

✓ تعداد دستور العمل ها نسبتا کم

✓ روش های آدرس دهی نسبتا کم

✓ دستیابی به حافظه منحصر به دستور بار کردن (load) و ذخیره سازی (store) است.

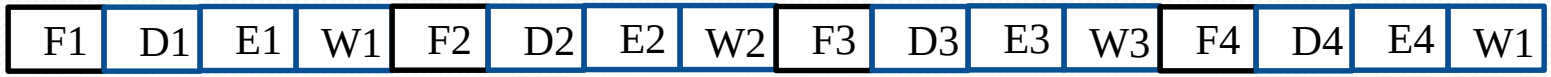
✓ تمامی عملیات در داخل ثبات های CPU انجام می شود.

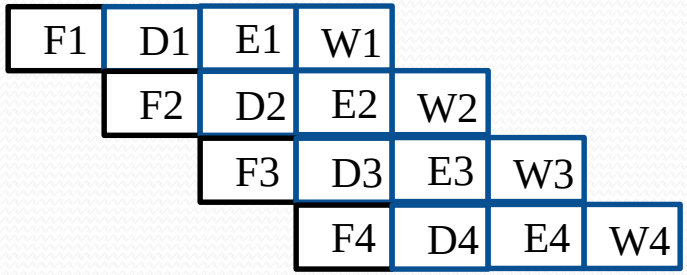
✓ دستورات طول ثابتی دارند و بسادگی دیکد می شوند.

✓ اجرای دستورات در یک سیکل انجام می شود.

تکنیک خط لوله ای (Pipeline)

- روشی برای اجرای سریع تر عملیات

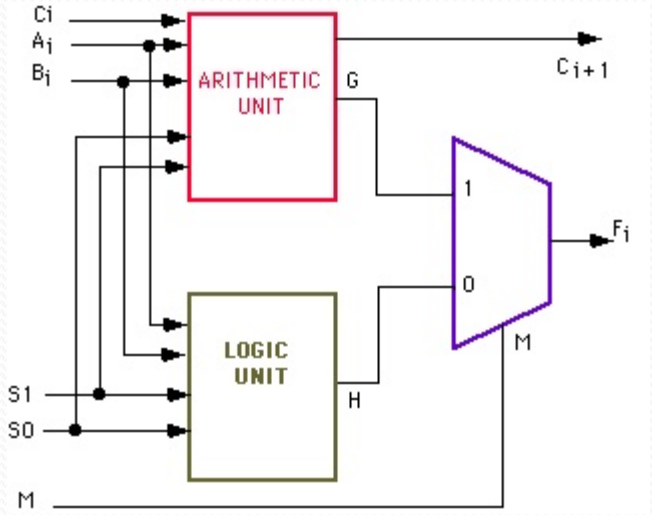




علت کاربرد بیشتر معماری Risc

- مجموعه دستورات کاهش یافته (امکان Pipelining)
- تعداد ثبات های بیشتر نسبت به معماری CISC

واحد محاسبه و منطق (ALU)

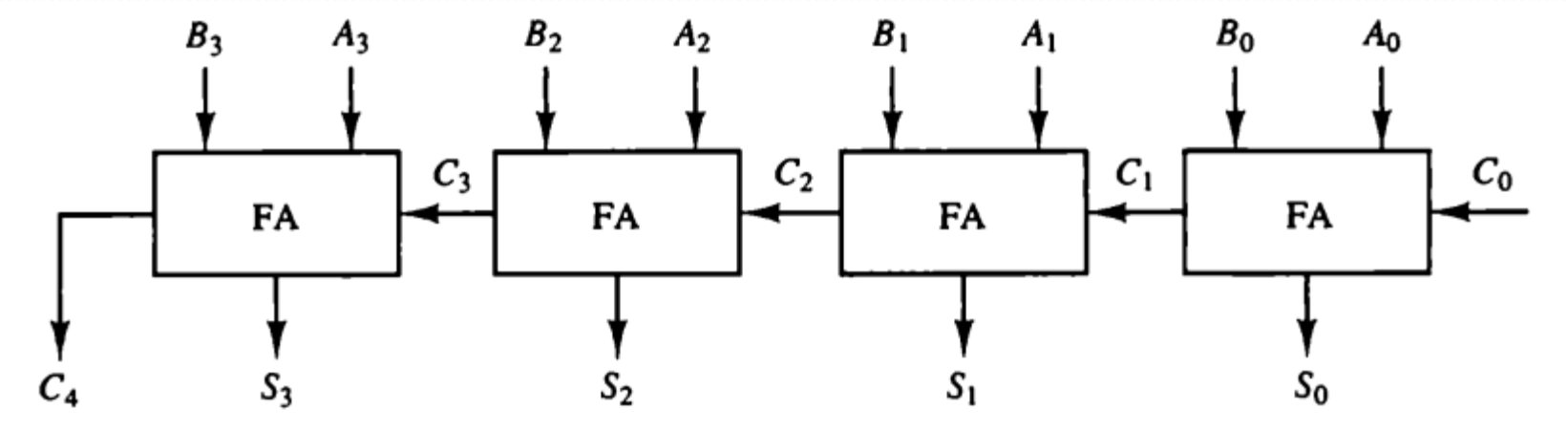


- ALU یک مدار دیجیتال است که عملیات محاسباتی و منطقی را اجرا می نماید.

واحد محاسبات

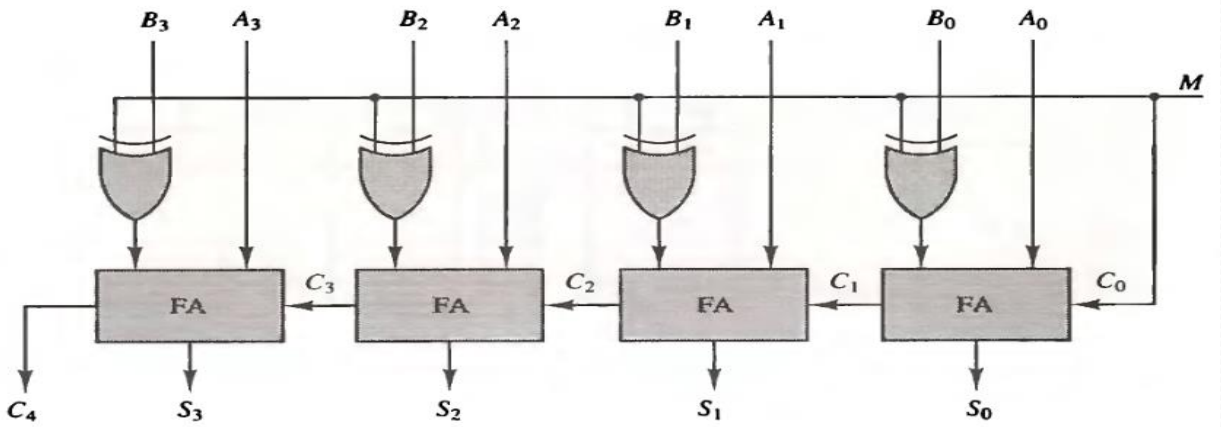
مثال ها:

جمع/تفریق/مکمل ۲/افزایش/کاهش

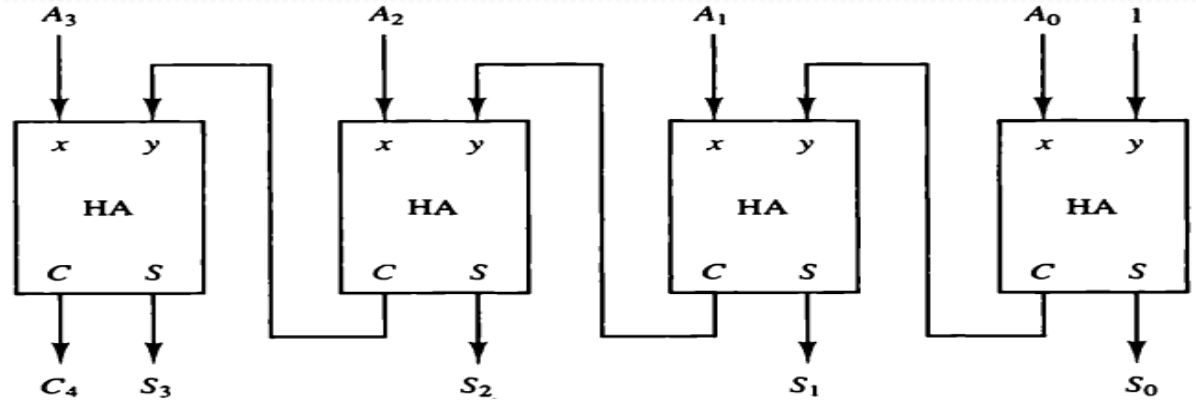


جمع کننده ۴ بیتی

واحد محاسبات



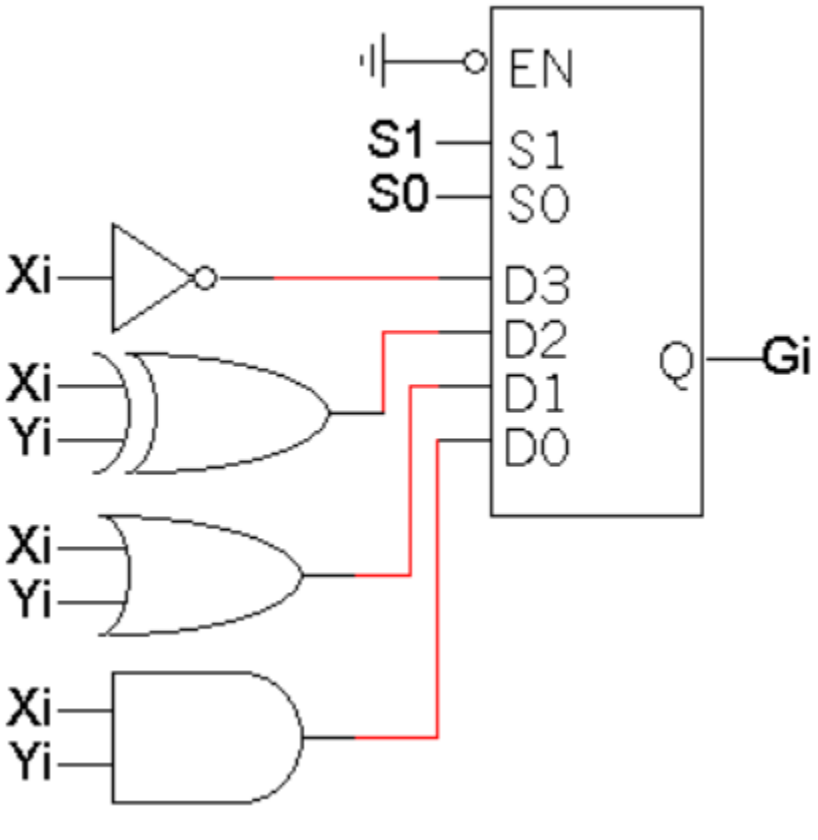
تفریق کننده ۴ بیتی



افزایش دهنده ۴ بیتی

واحد منطق

مثال:

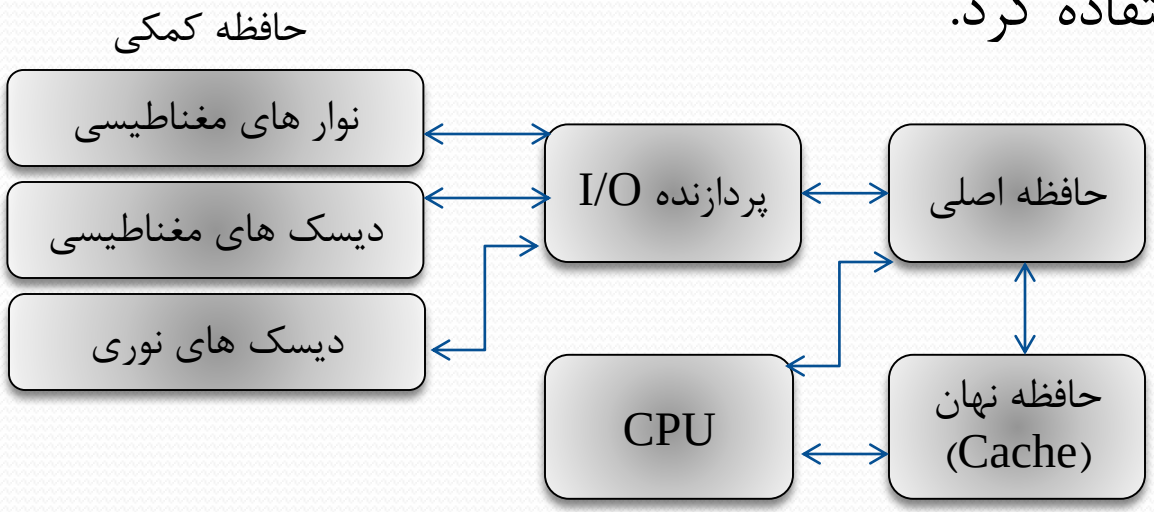


S_1	S_0	Output
0	0	$G_i = X_i Y_i$
0	1	$G_i = X_i + Y_i$
1	0	$G_i = X_i \oplus Y_i$
1	1	$G_i = X_i'$

حافظه

تعریف: سلول های دیجیتالی که می توان در آنها اطلاعات باینری را ثبت کرده و در موقع لزوم از آنها استفاده کرد.

سلسله مراتب حافظه:



انواع حافظه ها از نظر کاربرد:

- ✓ حافظه های فقط خواندنی (مانند: ROM)
- ✓ حافظه با قابلیت خواندن و نوشتن (مانند: RAM)

انواع حافظه

حافظه ROM و مورد استفاده

1) Mask ROM: توسط کارخانه مقدار دهی می شود.
2) PROM (Programmable ROM): توسط کاربر فقط یکبار مقدار دهی می شود.
3) EPROM (Eraseable ROM): با استفاده از اشعه UV اطلاعات پاک می شود.
4) EEPROM یا E2PROM (Electrically EPROM): با استفاده از سیگنال الکتریکی اطلاعات پاک می شود.
5) Flash EPROM: اطلاعات بصورت دفعی پاک می شود.

مورد استفاده: ذخیره سازی برنامه های راه اندازی، برنامه سرویس دهنده دستگاه های ورودی-خروجی (BIOS)، کد های برنامه و اطلاعات ثابت مثل کد اسکی

! توجه: در این نوع از حافظه ها داده ها با قطع تغذیه پاک نمی شوند.

انواع حافظه

تفاوت E2PROM با Flash

- E2PROM قادر است تا برروی یک بایت از حافظه اطلاعات را بنویسد ولی این وضعیت در Flash memory ها با نوشتن بر روی بخشی از حافظه (sector) که آن بایت را در خود دارد، صورت می پذیرد.
- معمولا به علت گران بودن آنها حداکثر تا ۱۲۸ کیلو بایت ساخته می شوند.

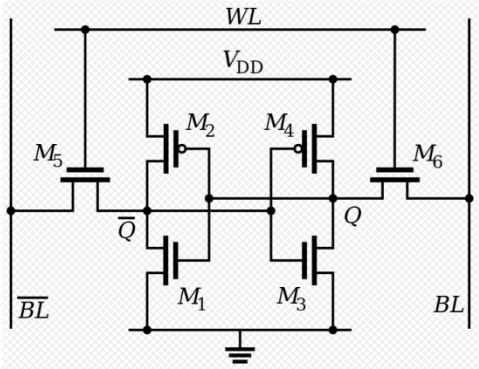
انواع حافظه

RAM

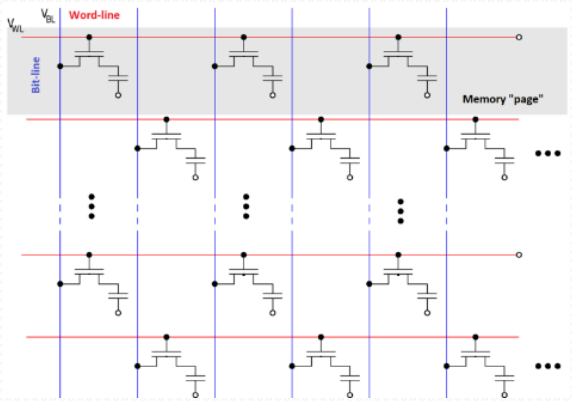
- Random access memory :RAM
- حافظه با دستیابی تصادفی که در صورت قطع تغذیه اطلاعات آن پاک می شود.
- این حافظه اجازه می دهد تا اطلاعات با زمان یکسان صرف نظر از مکان فیزیکی داده در درون حافظه خوانده یا نوشته شود.
- در دیگر حافظه ها مانند دیسک های نوری و مغناطیسی مکان فیزیکی داده اهمیت دارد.
- در حافظه های فلش دستیابی تصادفی به خواندن وجود دارد اما در نوشتن این امر امکان پذیر نیست.
- موارد استفاده از RAM: تعریف متغیر ها و داده ها، پشته

انواع RAM

- SRAM: مجموعه ای از DFF هاست و هر بیت داده توسط ۶ ترانزیستور ذخیره می شود.
- Hold :WL=‘0’
- Read/Write :WL=‘1’

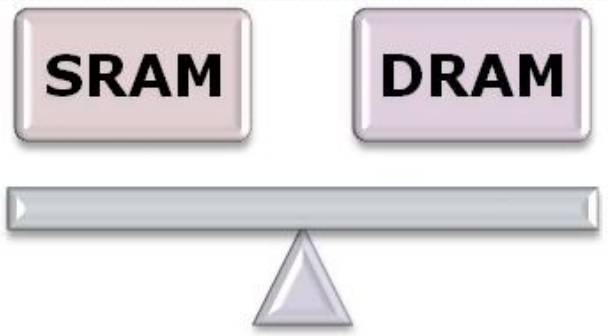


- DRAM: هر بیت داده توسط یک خازن کوچک و یک ترانزیستور ذخیره می شود.

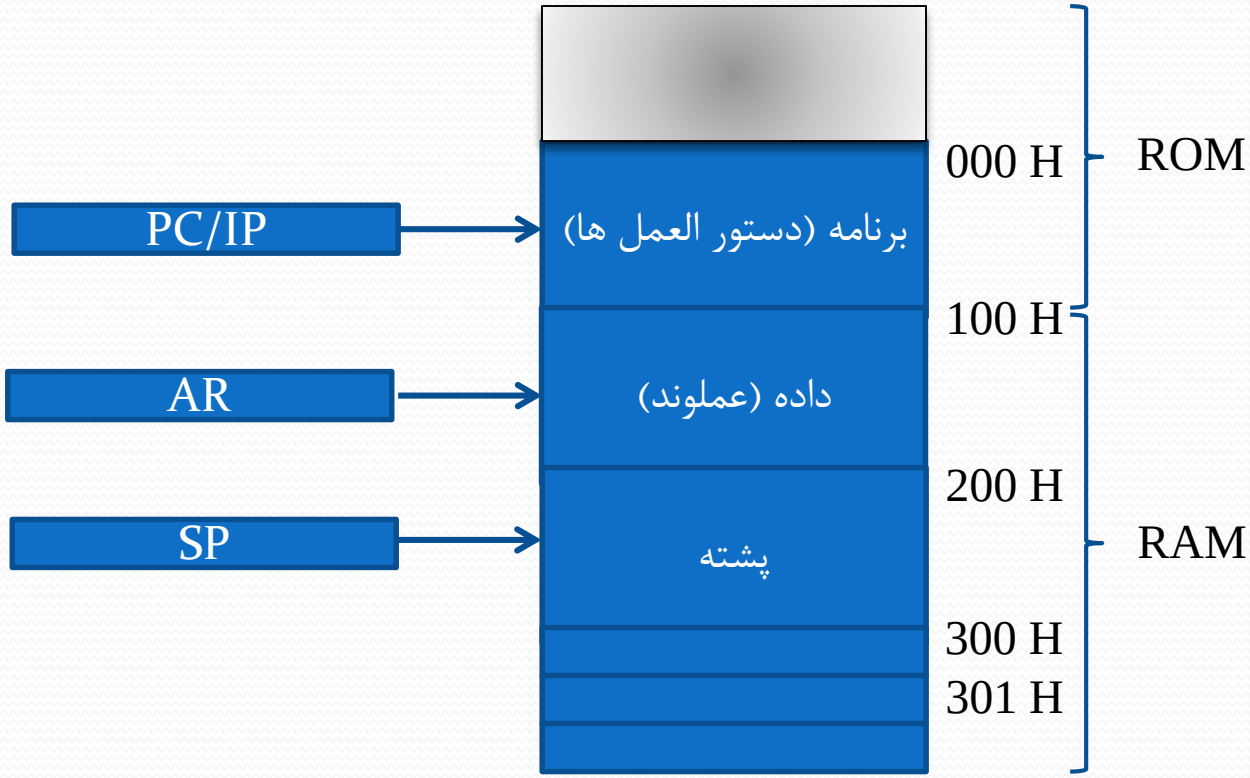


مقایسه SRAM با DRAM

- SRAM از DRAM سریع تر است و لذا از آنها جهت پیاده سازی حافظه نهان (Cache) استفاده می شود.
- DRAM دارای چگالی بیشتر است و لذا از آن جهت پیاده سازی حافظه اصلی استفاده می شود.
- به علت وجود جریان نشتی در DRAM، از مدارات جانبی جهت رفرش کردن اطلاعات (چند میلی ثانیه) استفاده می شود. لذا توان مصرفی بالاتری دارند.
- تعداد ترانزیستور کمتر در یک سلول حافظه DRAM منجر به کاهش هزینه تولید آن در مقیاس بالا می گردد.

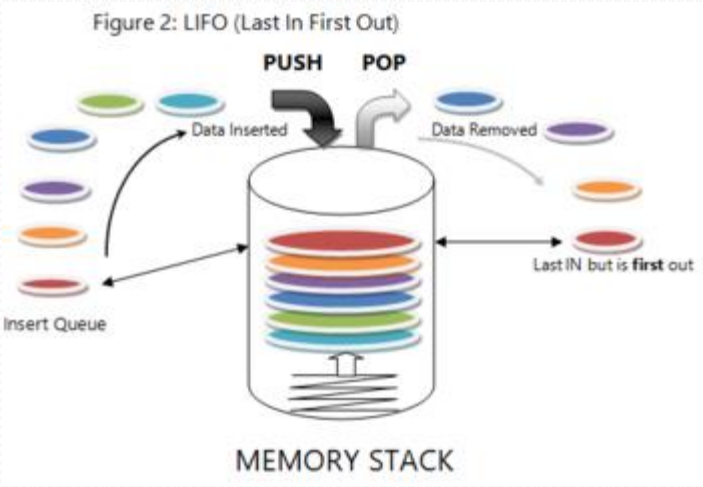


حافظه و نحوه آدرس دهی

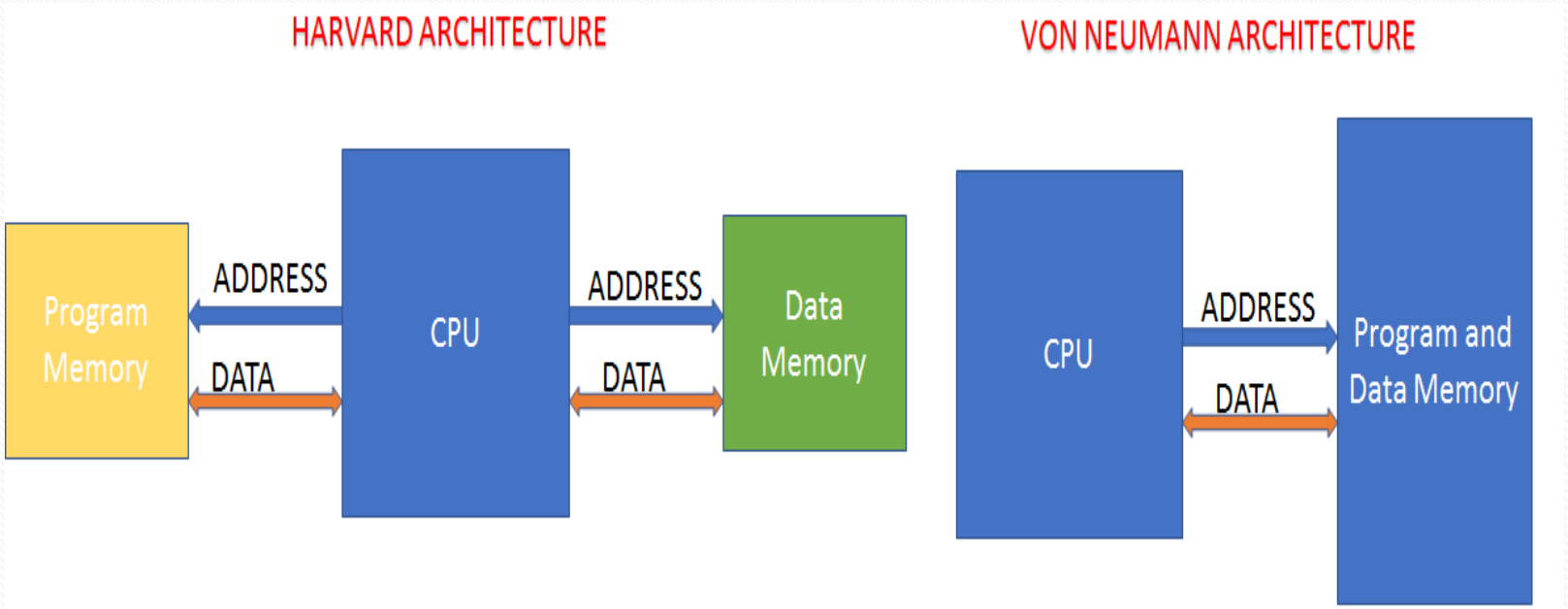


حافظه پشته

- فضایی در حافظه جهت ذخیره موقت اطلاعات
- بلوک داده در این حافظه بصورت LIFO است.
- ثبات SP (اشاره گر پشته) به بالای پشته اشاره می کند.



معماری ها جهت دسترسی به حافظه



ورودی-خروجی ها (I/O)

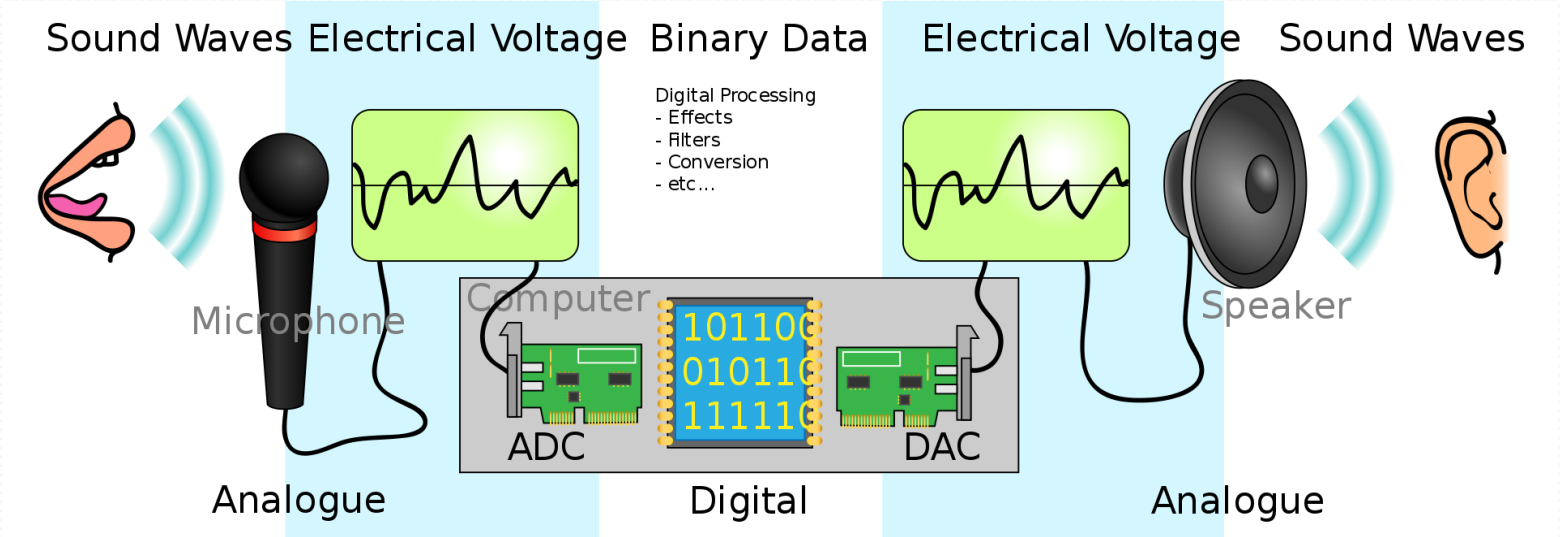
انواع کمیت ها:

فیزیکی: دما، رطوبت، نیرو، فشار، فاصله، سرعت، شدت نور/صوت

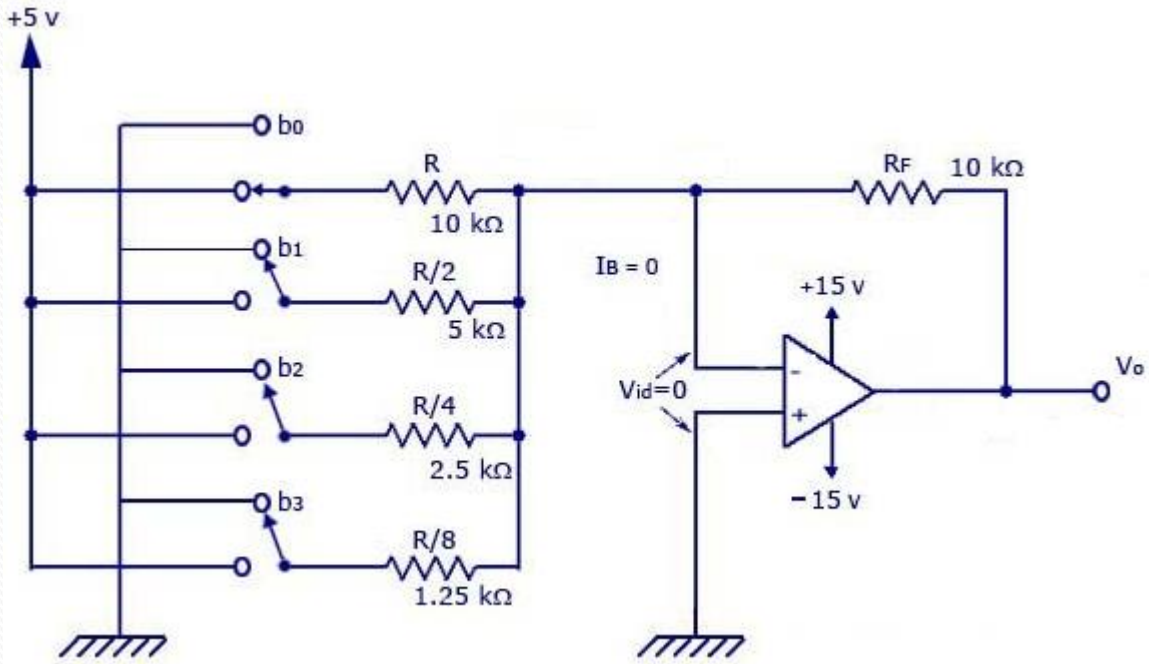
الکتریکی: ولتاژ، جریان، بار، شار

انواع ورودی ها: دیجیتال و آنالوگ

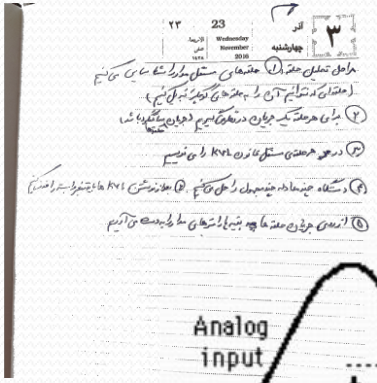
انواع خروجی ها: دیجیتال و آنالوگ

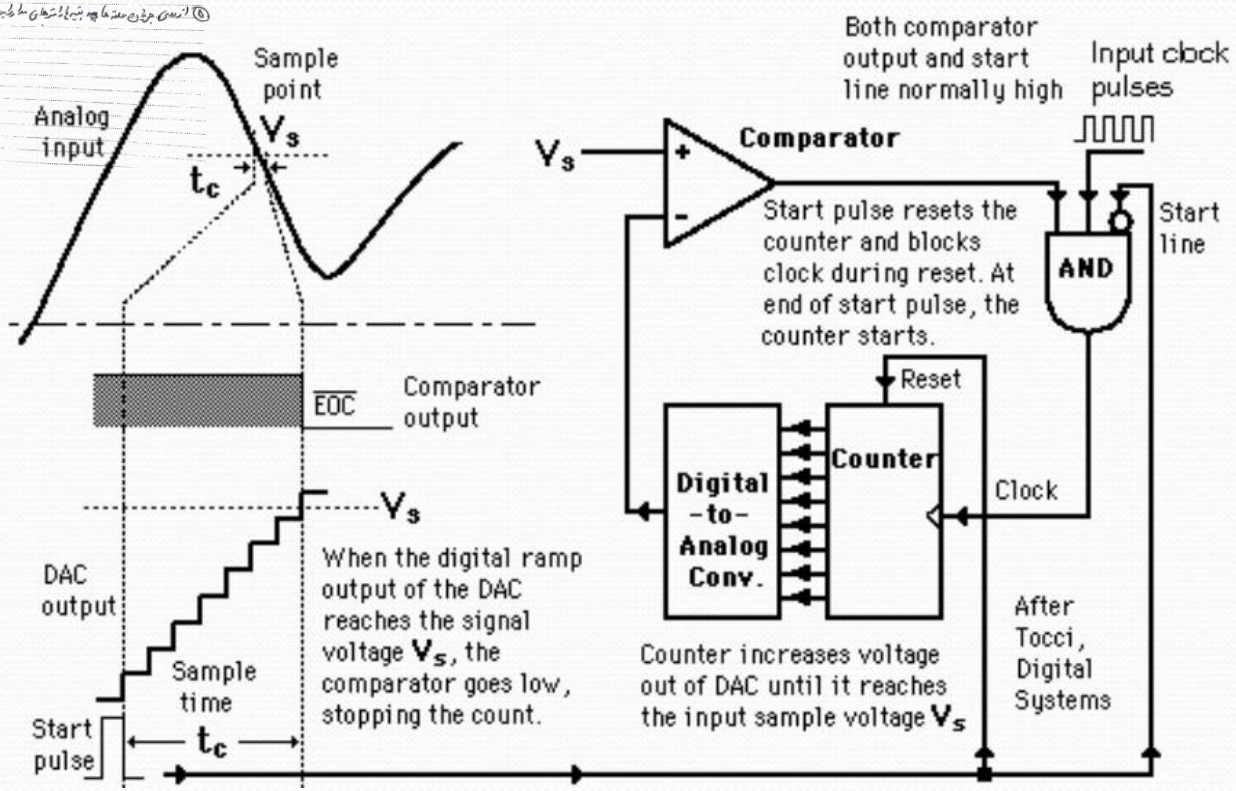


مبدل دیجیتال به آنالوگ (DAC)



مبدل آنالوگ به دیجیتال (ADC)





انواع CPU

- Intel
  - ✓ Celeron
  - ✓ Pentium
  - ✓ Xeon
  - ✓ Core/Core i

- AMD
  - ✓ Sempron
  - ✓ Athlon
  - ✓ Phenom

انواع CPU

- Intel

4-bit processors: 4004 (1974, 740kHz)

8-bit processors: 8008/8080/8085

16-bit processors: 8086/8088/80186/80188/80286

32-bit processors: 80386 DX(SX)/80486 DX(SX)

Pentiums (II, III, IV)/Celerons/Xeons/Intel core/ Dual core Xeon

64-bit processors: Itanium/Pentium (4F)/Xeon/Intel core 2/Celeron/Core i3/i5/i7

- AMD

4-bit processors: AM2900/AM2901

32-bit processors: AM29000/AMX86: 3, 4, 5

64-bit processors: Athlon/Sempron/Phenom FX:X4:X3

معیار ها جهت تشخیص سرعت CPU

- عوامل تاثیرگذار در سرعت:

✓فرکانس

✓تعداد هسته

✓میزان حافظه Cache

شاخص ها:

- Millions of instructions per second (MIPS)
- Floating point operations per second (FLOPS)

انتخاب تراشه ها برای پردازش بر اساس نوع داده ورودی