سه‌شنبه ۱۳ـ۱۲

Register Transfer Language (RTL) :

توصیفی است از انتقال ثبات‌ها .

- هزینه‌ی FF ۲٫۵ برابر یک Latch می‌باشد .

• داوری به ازای سرعت و هزینه :

PPC → Performance per Cost : $\frac{\text{Performance}}{\text{Cost}}$ $\left(\frac{\text{سرعت}}{\$}\right)$

$\$ \propto$ gate

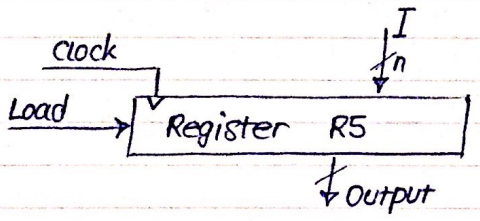
- سرعت FF دو برابر کندتر است .

- با وجود این FF به خاطر کنترل بهتر به Latch ترجیح داده می‌شود .

FF $\xrightarrow{\text{Types}}$ S_R → J_K → T → D

D_FF with Load (Enable) → کنترل بهتر

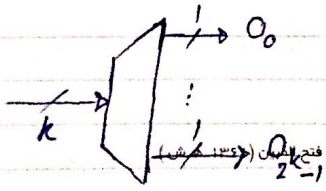
وقتی به هم بسته شوند :



MUX , DeMUX

Decoder , Encoder

Decoder :



تنها جایی که نرم‌افزار تبدیل به سخت‌افزار می‌شود .

عید نوروز (تعطیل) – هجوم ماموران ستم شاهی پهلوی به مدرسه فیضیه قم(۱۳۴۲ هـ ش) – آغاز عملیات فتح‌المبین (۱۳۶۱ هـ ش)
به‌نخش فراصرفه پاکسان ایران مروارید سلمون سبزینا گلناشت پاکسان

Encoder همانند MUX است بدون خط کنترل (تبدیل سخت‌افزار به نرم‌افزار)

→ Priority Encoder

در DeMUX به جز یکی از خروجی‌ها بقیه Z می‌باشد. ← اشکال

؟ می‌خواهیم به سطح و می‌خواهیم به لبه؟

موافقتی که برای ما مهم نیست پس در خروجی داده را ذخیره کند ← سطح

عید نوروز (تعطیل)

به‌پخش فراصرفه آی‌تی‌کو یاکسان ایران مروارید سلمون ساجنا گلشاد یاکسان

سلسله مراتب حافظه (Memory Hierachy) :

Top to Down → CPU داخل (on chip)

CPU ⇄ t_1 FF → Register → Register File → Cache

→ Main Memory → Magnetic Disk → Optical Disc

→ Tape / punched card t_8

← سرعت بالا (تأخیر نوشتن و خواندن پایین)

→ کاهش هزینه

→ افزایش ظرفیت

t_i : تأخیر → $t_1 < t_2 < \dots < t_8$

c_i : هزینه هر بیت → $c_1 > c_2 > \dots > c_8$

v_i : ظرفیت → $v_1 < v_2 < \dots < v_8$

h_i : hit rate = $\frac{\text{تعداد دفعات موفق بودن}}{\text{تعداد کل دفعات درخواستی}}$ → $h_1 < h_2 < \dots < h_8$

$h_1 \simeq \varepsilon \quad h_n = 1$

برای ساخت یک CPU : سرعت ↑ قیمت ↓ حجم ↑

$\text{Min}\left[\text{Cost} = \sum x_i c_i\right]$ $\quad x_i$: تعداد بیت

$\text{Max}\left[v = \sum x_i\right]$

تقریبی :

$$\text{Average Access time} = h_1 t_1 + (1-h_1)\{h_2 t_2 + (1-h_2)\{h_3 t_3 + \dots\}\}$$

پاکدیان گلشاد ساجنا مروارید سامون پاکدیان ایروان آیکو فرامرفه به‌پخش

OR $= h_1 t_1 + (1-h_1)\{t_1 + h_2 t_2 + (1-h_2)\{t_2 + h_3 t_3 + \cdots\}\}$

$= t_1 + (1-h_1)\{t_2 + (1-h_2)\{t_3 + \cdots\}\}$

$\Rightarrow$ Min (Average Access time)

هر طبقه‌ای miss بخورد از طبقه پایین‌تر پیدا می‌کند.

hit rate + miss rate = 1

* پراستفاده‌ها در طبقه‌های بالاتر

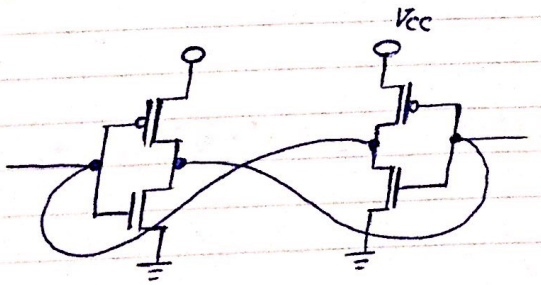
Register File ← چند Register در کنار هم

• Main Memory (RAM)

Memory
- Direct Access → RAM, ROM
- Sequential Access → Tape

RAM
- DRAM (Dynamic RAM) ⇒ ارزانی، لوحی (خیال) — Refresh ← به مرور زمان دشارژ می‌شود
- SRAM (Static RAM) — خودش را Refresh می‌کند.

ترانزیستور ← سریع و بزرگ



• حافظه‌ها :

۱- آدرس‌پذیر :

A : دسترسی مستقیم ←

I : RAM

- SRAM
- DRAM

II : ROM

- ROM
- PROM
- EPROM
- EEPROM

B : دسترسی ترتیبی ←

- Tape

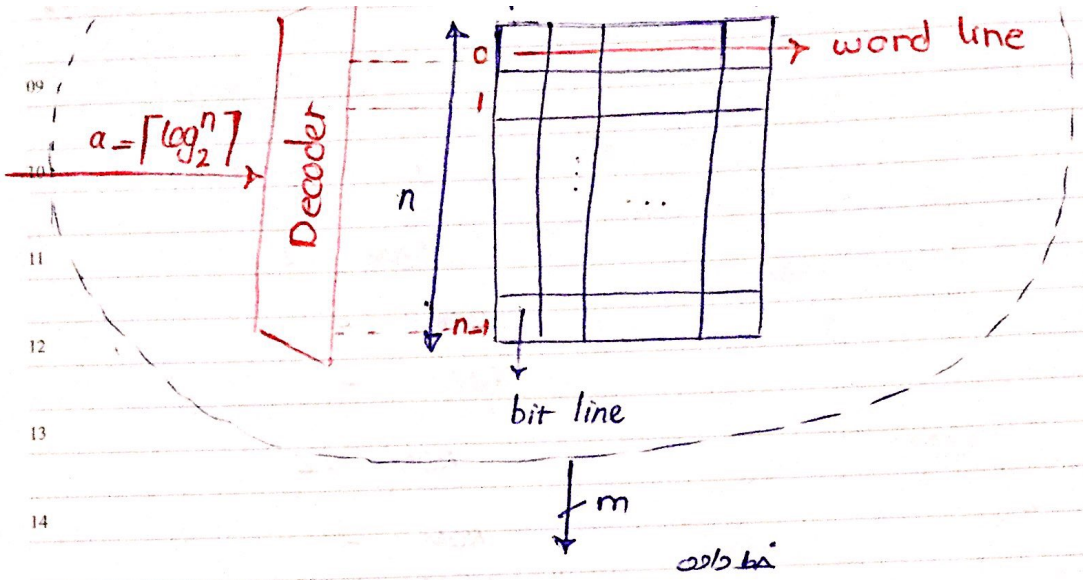
و ...

۲- آدرس‌ناپذیر :

CAM ← Content Addressable Memory

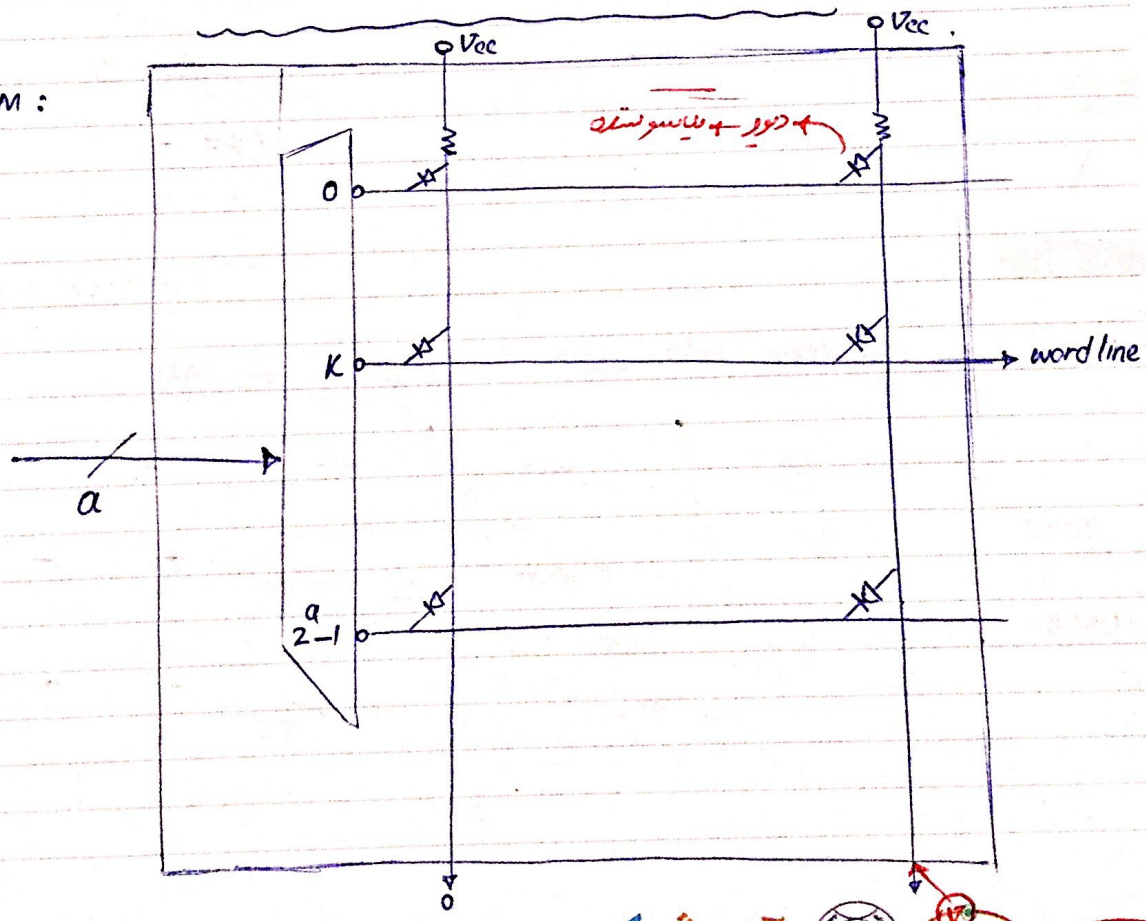
	حافظه	معایب	مزایا	سرعت	توان مصرفی	مساحت
SRAM	ایستا	گران مساحت زیاد	سرعت بالا توان مصرفی پایین	+	−	+
DRAM	پویا	نیاز به Refresh توان مصرفی بالا	ارزان پرظرفیت	−	+	−

به‌پخش فرامرفه آیروان مروارید سایمون سایجا گلشاد یاکسان



$$\text{حجم} = \frac{m \times n \ (\text{بیت})}{8} \ : \ \text{بایت}$$

ROM :



به‌نیش
فراصرفه
یاکدیان ایران
مروارید
ساجنا
گلتاش
یاکدیان

برای این که ! شود ، دود را می‌سوزانند .

DRAM ، SRAM هیچ لام جایگزین دیگری نشده است .

DDR → Double Data Rate

SDRAM ، DDR ها از خانواده‌ی DRAM ها هستند .

داخل تراشه ، سرعت بیش‌تر ← SRAM

CAM → Data آدرس پذیری بر اساس

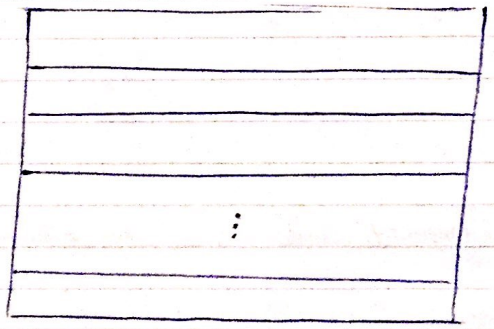
بخشی از اطلاعات را می‌دهیم که سه حالت جواب دارد :

۱) نسبت . ۲) یک داده وجود دارد . ۳) چند داده وجود دارد .

اگر ! باشد مقصر است .

MASK ○ ○ ○ ... ○

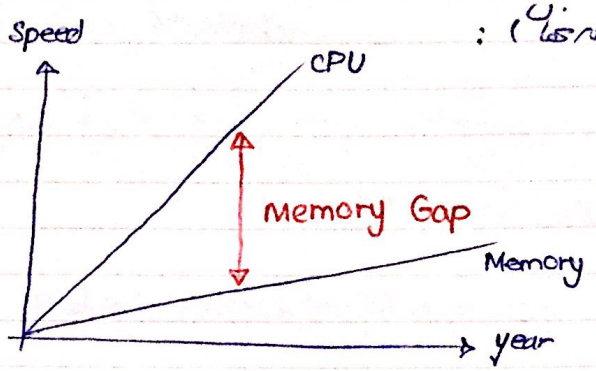
DATA ○ ○ ○ ... ○



مصرف برق و توان بسیار زیادی دارد .

به‌نخش فرامرفه آی‌تی‌سی یاکسیان ایروان مروارید هلمون ساجنا گلناشت یاکسیان

. Cache (حافظه نهان) :

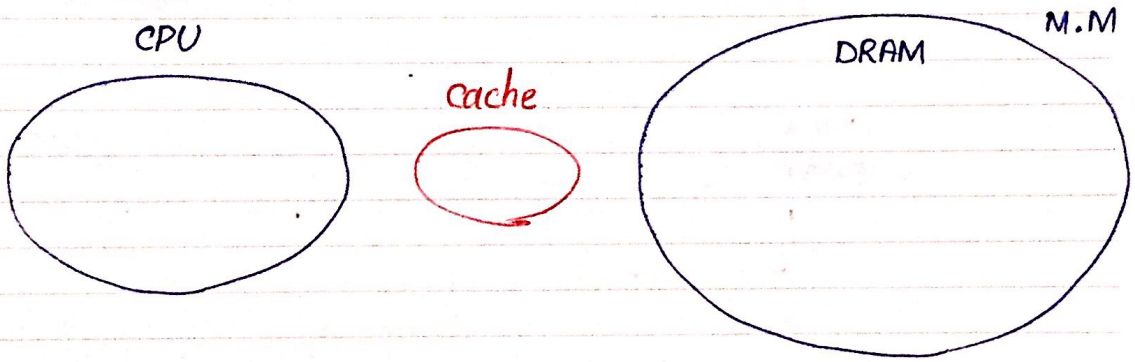


CPU دارای clock

M.M ← بدون clock

cache از نوع SRAM است.

می‌خواهیم حافظه‌ای سریع و ارزان داشته باشیم به همه‌ی داده‌های M.M را داشته باشد.



به خاطر پیشرفت کند حافظه اصلی آن را Memory wall می‌نامند.

• چنان چه حافظه اصلی تأخیر 10^{-6} داشته باشد، اگر از حافظه نهان با تأخیر 10^{-9} و نرخ برخورد 95% استفاده شود، مطلوب است تأخیر متوسط دسترسی به حافظه.

$$T_{Access\ Time} = h_1 t_1 + (1-h_1)(t_1+t_2)$$

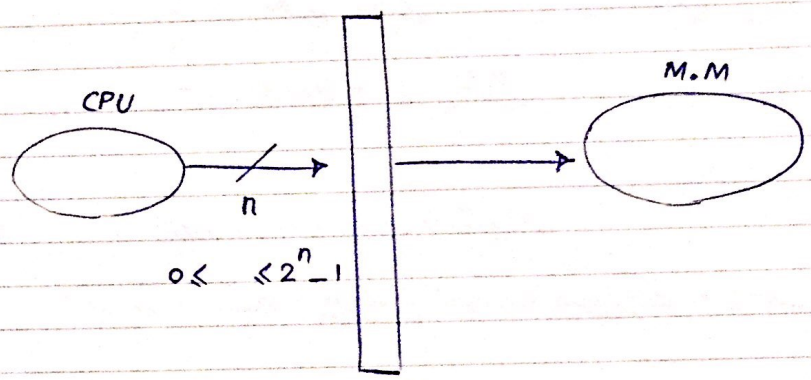
$$= 0.95 \times 10^{-9} + 0.05(10^{-9}+10^{-6}) \simeq 0.51 \times 10^{-9}\ (s)$$

$$\text{افزایش سرعت} = \frac{\text{تأخیر کند}}{\text{تأخیر تند}} = \frac{10^{-6}}{51 \times 10^{-9}} \simeq 20$$

به‌نشر فرامرفه آی‌تی‌کده ایران مروارید سامون ساجنا گلتاش پاکسان

• وجود cache به خاطر کنار هم قرار گرفتن CPU و M.M است. به همین دلیل در داخل M.M از cache استفاده نمی‌کند (هم چنین به دلیل گرانی).

• اگر ۱: همجواری مکانی Spatial Locality ۲: همجواری زمانی Temporal Locality وجود داشته باشد، وجود cache سرعت را افزایش می‌دهد.



آدرس را در M.M می‌دهد.

همجواری زمانی

آدرس‌ها → 3, 5, 4, 10, 5, 1000, 1200, 1201, 3, 5, 8

همجواری مکانی

به طور مثال خانه‌ی ۳ را در حافظه‌ی نهان می‌گذارد و از آن جایی که هزینه‌ای را عوض دارد می‌شود تا از M.M به cache بیاورد، مکان‌های نزدیک آن را هم قبلاً درخواست شده نیز در cache قرار می‌دهد (مثلاً ۵ و ۴)

مثال همجواری مکانی ← آرایه‌ها

```
int a,b;
a=1;
b=1;
a=a+b;
```

100	Data
101	
[100] ← 1	Instruction
[101] ← 1	
[100] ← [100] + [101]	

همجواری مکانی دارند.

به‌نشر فرا صرفه یاکسان ایران مروارید سایبان گلتاش یاکسان

• اگر همجواری مکانی را از بین ببریم، ارزش cache از بین می‌رود.

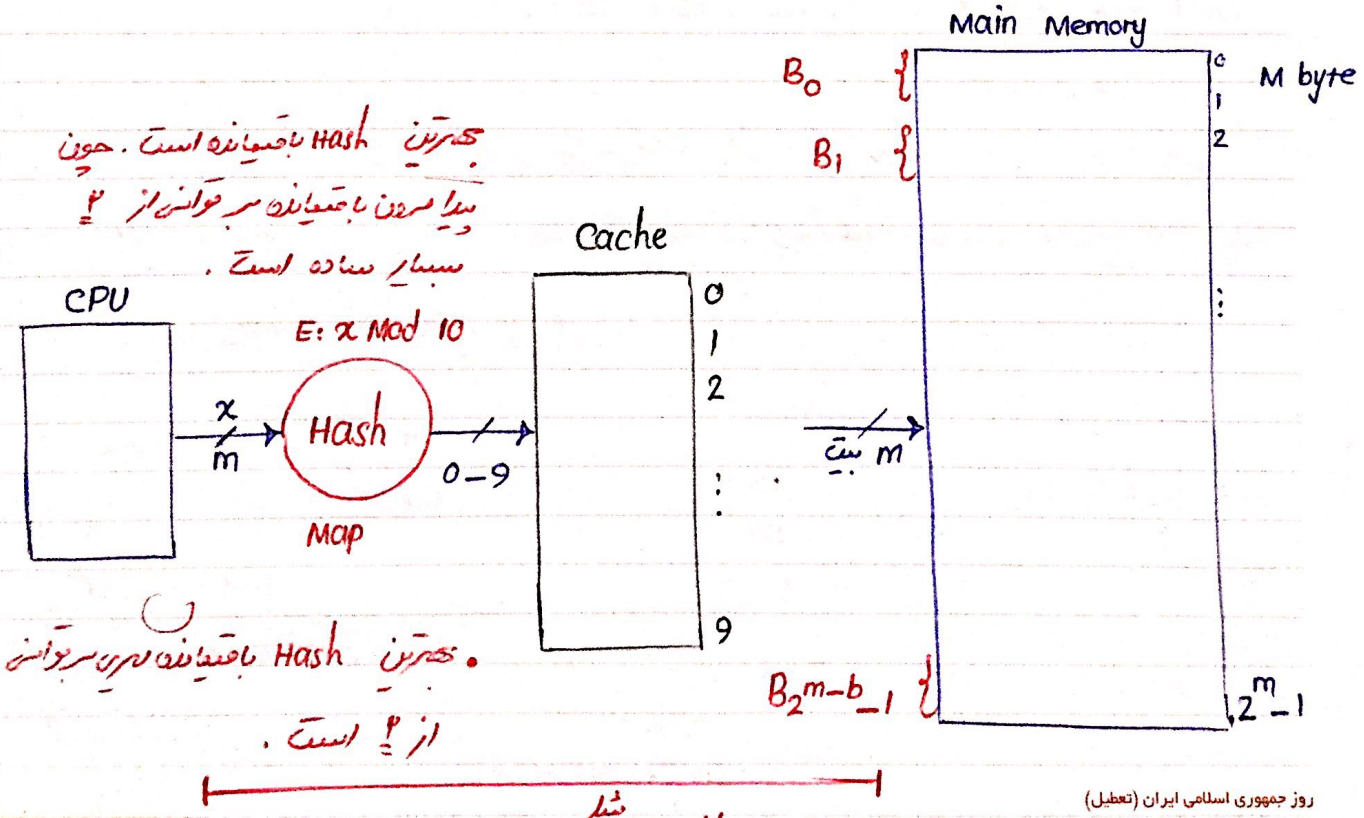
مثال‌هایی از همجواری زمانی:

- متغیرهایی که چندین بار استفاده می‌شوند.
- حلقه‌های For
- goto
- Call Function

• Compiler می‌تواند همجواری مکانی و زمانی را تشدید کند.

• می‌توان به جای این که با استفاده از نرم‌افزار سرعت را افزایش داد، با سخت‌افزار این کار را انجام داد.

طراحی cache:



به‌نشر فرامرفه ایران آی‌تی یاکدیان ایروان مروارید سلمون ساجنا گلناشت یاکدیان

Direct Mapped Cache ← دیدگاه اول

حافظه نهان نگاشت مستقیم ٪۹۵,۵ $\xrightarrow{\text{پیشرفت}}$ ٪۹۵

مشکل : نگاشت شدن به یک خانه ← هرچی ↑ Cache مشکل ↓

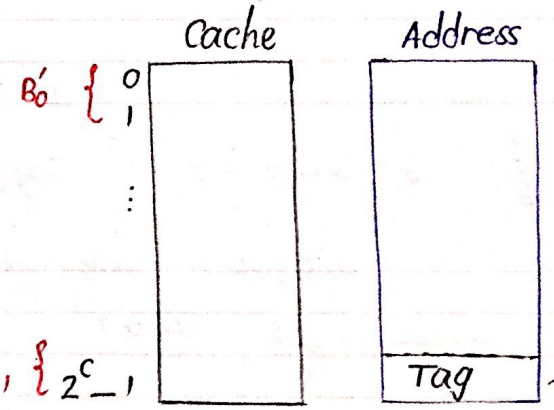
همواری زمانی ← نگه داشتن Data

درصورتی که بخواهیم همواری مکانی را رعایت کنیم ←

۱) مرده بزرگ باشد : زمانی از بین می‌رود.

۲) مرده کوچک باشد : مکانی از بین می‌رود.

حافظه Cache به خاطر باقیمانده‌گیری ساده‌تر ، توانی از ۲ است.



برای این که بدانیم خانه‌هایی که باقی مانده شان پر است ، کدام در Cache پر شده است.

آدرس اصلی ~ $\frac{\text{ظرفیت Cache}}{\text{محل در Address}}$

محل در Cache

Main Memory به Block تقسیم می‌کنیم.

Block $B = 2^b$

در این مثال $b=1$

به پخش فرا صرفه آی‌تی‌کی پاکسان ایران مروارید سامان ساجنا کتاباش پاکسان

هرچه اندازه Block بزرگتر باشد، مقدار عددی m باید بر آن باقی مانده بگیریم کم‌تر می‌شود. از این پس واحد صحبت‌های ما Block خواهد بود.

x : آدرس درخواستی داده از طرف CPU برحسب بایت

m آدرس درخواستی برحسب Block

$\frac{x}{B} \longrightarrow x_b$ آدرس برحسب Block : خارج قسمت

x_r : باقی‌مانده محل نسبی بایت در Block

محل در Cache m

$x_b \div \frac{C}{B} \longrightarrow$ خارج قسمت : Tag برای مشخص شدن

آدرس Block مورد نظر در Cache : باقی‌مانده

x_b $c-b$ بیت سمت راست (Index)

وقتی توانی از ۲ باشد ←

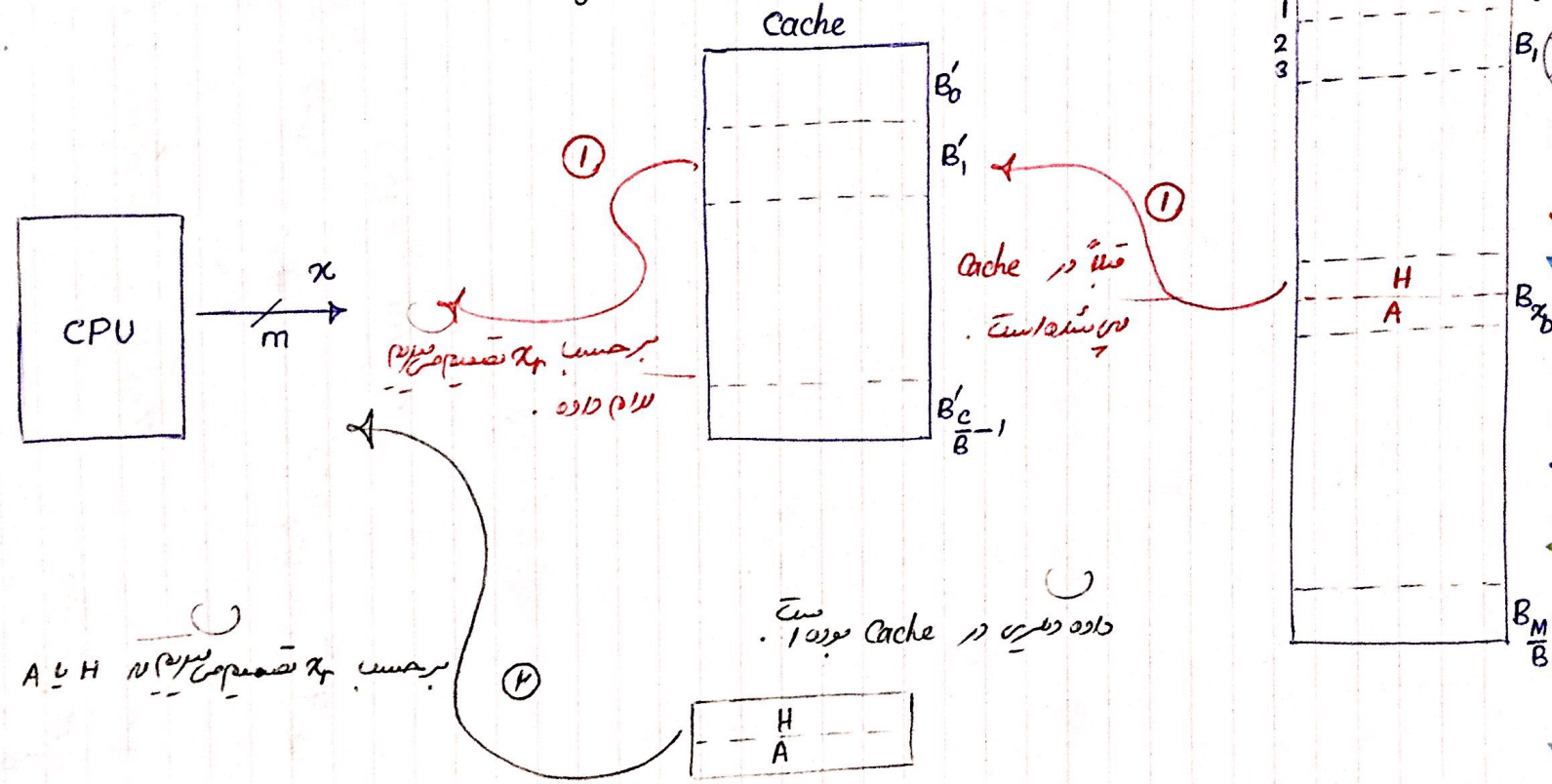
x_b : $m-b$ بیت سمت چپ

x_r : b بیت سمت راست.

$$\underbrace{(\text{خارج قسمت}}_{n-s} \mid \underbrace{\text{باقی‌مانده})_r}_{s} \div r^s$$

$x_b \div \frac{C}{B}$

- باقی‌مانده = ۱
- خارج‌قسمت
  - = Tag ①
  - ≠ Tag ②

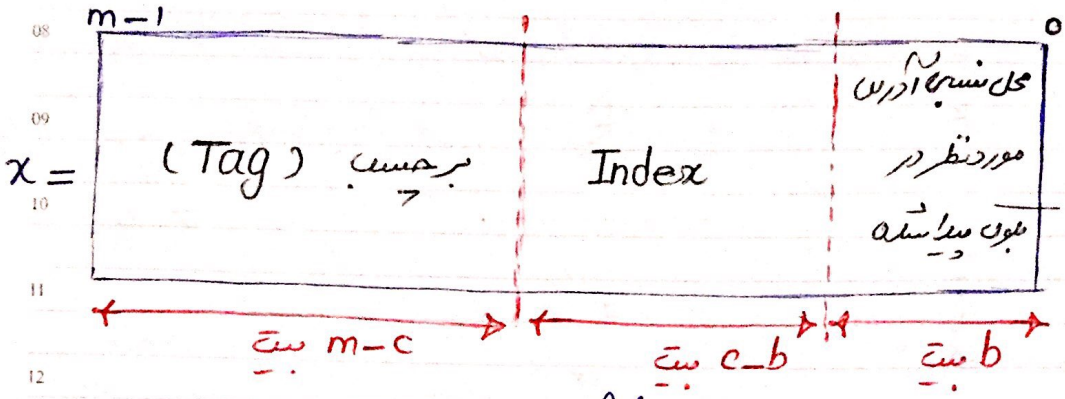


پس از خواندن از M.M در Cache وارد می‌شود.

یکشنبه ۱۳۹۴
۱۶
فروردین
SUNDAY
5 Apr 2015
۱۵ جمادی‌الثانی ۱۴۳۶

x_b : آدرس برحسب Block

x پس از بایتی شدن :



(قالب آدرس CPU)

Block بزرگی تعداد Tag ها را هم مرتبط .

به‌پخش فراصرفه آی‌تی‌کی ایران یاکسان مروارید سباجنا گلناش یاکسان

روز سلامتی (روز جهانی بهداشت)

به‌پخش
فرامرکز
آی‌تی‌کی
پاکسان ایران
مروارید
سامان
ساجنا
کتاشت
پاکسان

۵٪ به miss می‌خورد چیست؟ به خاطر خانه‌هایی هم باقی مانده. به خاطر همجواری زمانی در آینده تکرار می‌شود. به خاطر محدود بودن ظرفیت cache نمی‌توان به سادگی آن را حل کرد.

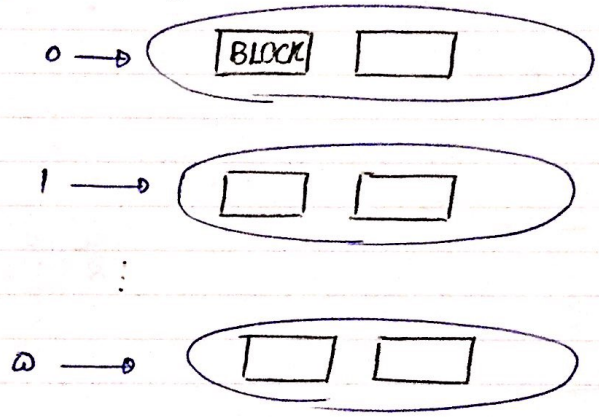
HOT BLOCK: خانه‌هایی که در cache خیلی به سراغ آن‌ها می‌رویم. ← در مقابل COLD B.

دیدگاه دوم: K-way Set Associative cache

(حافظه نهان مجموعه انجمنی k-تایی)

hit: 96 – 97 – 98 %

Associative ← در این خانه، هیچ لزوم برد خاصی ارجحیت ندارد.



⇒ ۱۵ تایی

2-way → بسیار پرکاربرد

MOD 10 ⇒ MOD 6

اجازه‌ی انتخاب بین k خانه را می‌دهد ← میزان hit افزایش می‌یابد.

در دیدگاه اول به تردد رفت و برگشتی توجهی نمی‌کنیم.

به‌نشر فرامرفه آرکی‌فی پاکسیان ایروان مروارید هلمون ساجد کتاباش پاکسیان

روز ملی فناوری هسته ای - روز هنر انقلاب اسلامی (سالروز شهادت سید مرتضی آوینی)(۱۳۷۲ هـ.ش)

LRU ، MRU به وسیله‌ی صف ساده پیاده سازی می‌شوند.

LRU: سمتی که hit می‌شود، به اول صف منتقل می‌شود. حذف شدن از اول صف ← MRU

از انتهای صف ← LRU

همگرایی رها دارند ولی همگرایی ما در M.M نمی‌نظرند.

① Placement Policy (سیاست جایابی)

② Replacement Policy (سیاست جایگزینی)

② →
- FIFO → بالا میزان hit را مشکل نیم، سی شکست می‌خورد. همانند فوق
- LIFO →
- LFU (فرکانس سنجی) least Frequently Used (از لحظه‌ی صفر)

MFU
LRU (Least Recently Used) (اخیراً)
MRU
Random

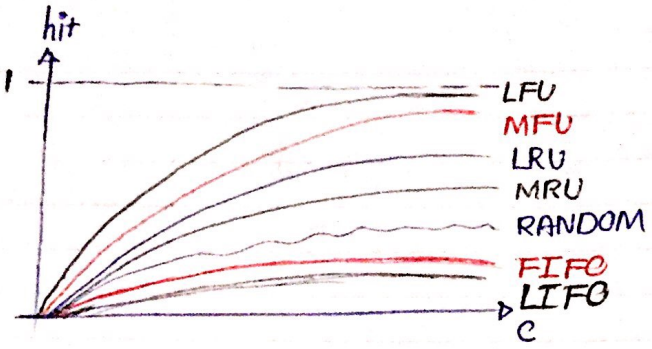
شهادت امیر سپهبد علی صیاد شیرازی (۱۳۷۸ هـ.ش)- ولادت حضرت فاطمه زهرا(س)(هشتم قبل از هجرت)و روز زن- تولد حضرت امام خمینی(ره)رهبرکبیر انقلاب اسلامی (۱۳۲۰ هـ.ق)-سالروز افتتاح حساب شماره ۱۰۰ به فرمان حضرت امام(ره)و تاسیس بنیاد مسکن انقلاب اسلامی (۱۳۵۸ هـ.ش)

به نخش
غذا صرفه
پاکسان ایروان
مروارید
سامان
ساجا
کشاورز
پاکسان

خطای LFU به تراست ولی نیاز به حافظه بیشتری دارد. MFU , LFU

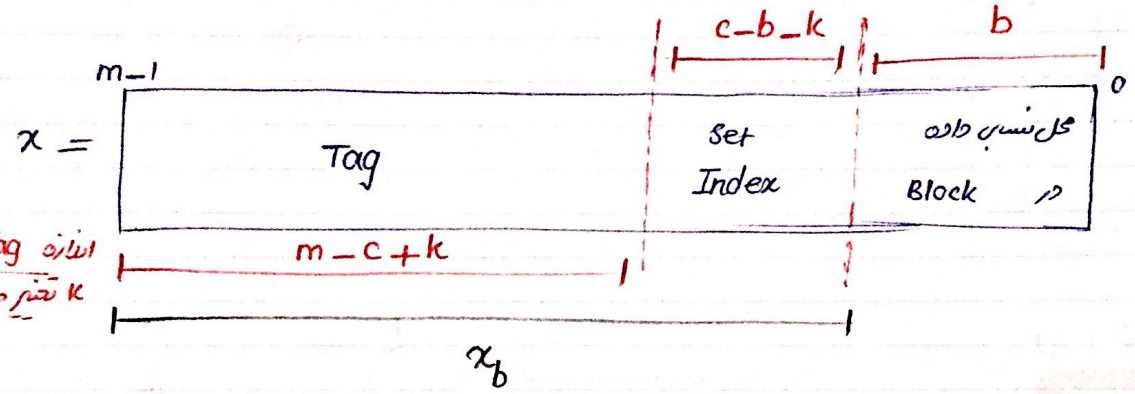
به خاطر مشکل نکته افزایش (counter)

قابل پیاده سازی نیست.



$k=4$

آدرس داده درخواستی از طرف CPU :



تأخیر در Set های با k بزرگ، زیاد است.

k اتلاف انرژی در مقایسه‌ی Tagها (اتصال به Buffer)

k_compare به صورت موازی (توان مصرفی زیاد XOR)

به خاطر بزرگی هنگام پی در Cache، hit بیش از یک خواهد بود.

پیاده‌سازی LRU (خط DeMUX) برای Replacement

↑ k_way Hit Ratio (a)

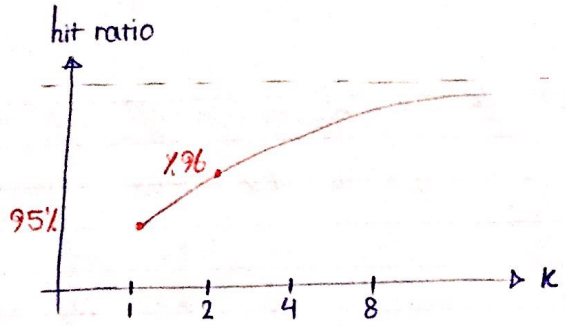
توان مصرفی ↑ k_way

(مدار) هزینه ↑ k_way

Delay ↑ k_way

K↑ ← بهتر

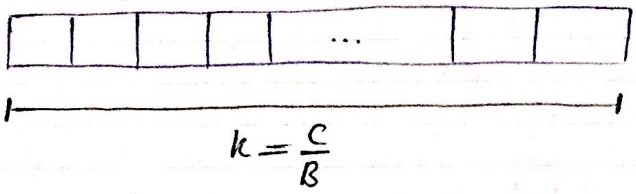
به‌نشر فرامرفه آی‌تی پاکسان ایران مروارید سامون سایبا کتاب پاکدیان



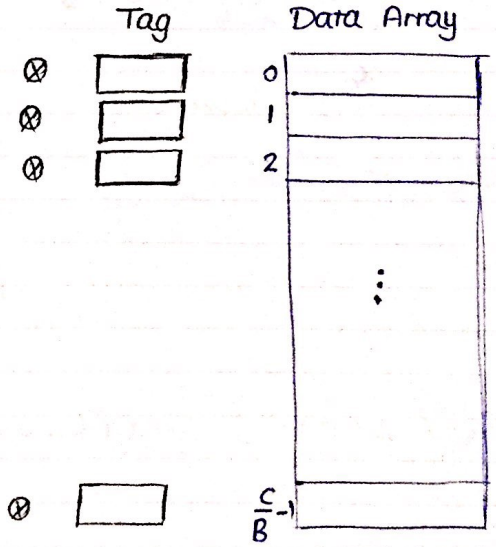
افزایش Delay خیلی چشمگیر نیست.

hit ratio نسبت به زمان ارجح‌تر است.

دیدگاه سوم: در تمام انجمنی Fully Associative Cache " hit : 99%



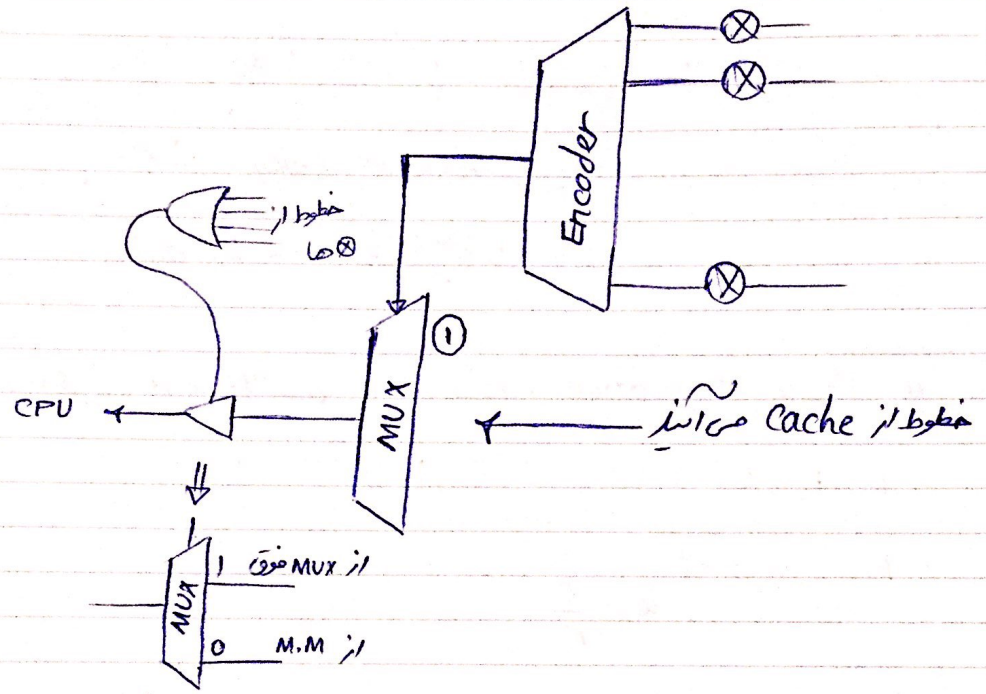
- Placement ندارد.
- نقش Replacement در آن بسیار پررنگ است.
- می‌توان ۹۰٪ آن را چنان تا همانند یک Cache معمولی باشد با این تفاوت که یک set هستند.



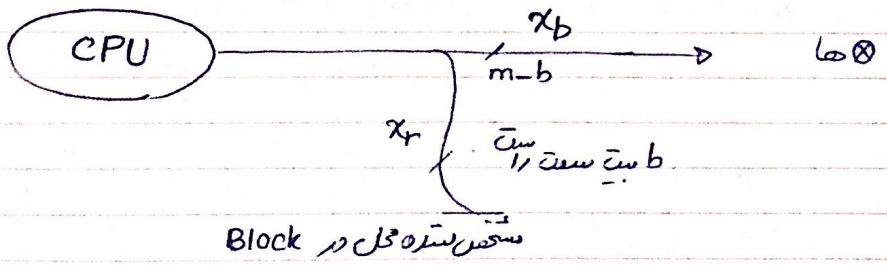
- Decoder ندارد.
- تعداد comparator ها بسیار زیاد است، به تعداد k.
- comparator ها موازی با هم کار می‌کنند.

به‌نیش فراصفحه آیکی پاکسان ایران مروارید همون سایت کتاباش پاکسان

هنگامی که به دنبال داده ای می‌گردیم، برای این که داده‌ی پیدا شده به CPU بازگردانده شود،
به یک Encoder و MUX k تایی نیاز داریم.



در Cache موجود است؟

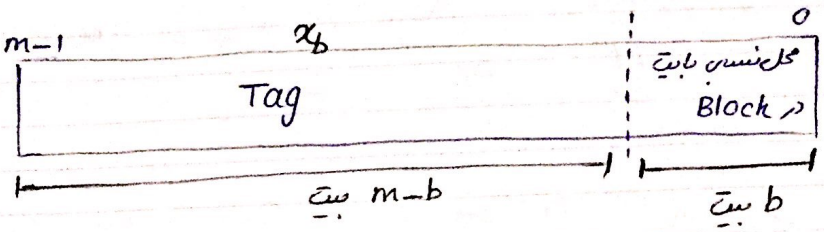


- حجم ① MUX بسیار زیاد است.

- Area (Fully. A. C) > Area (k_way. A. C) > Area (D.M.C)

به‌نشر فرامرفه آیکی ایران پاکسیان مروارید سلمون ساجد کتاشی پاکسیان

- پیدا کردن max در بین k سطر ملاز بسیار بزرگی لازم دارد.



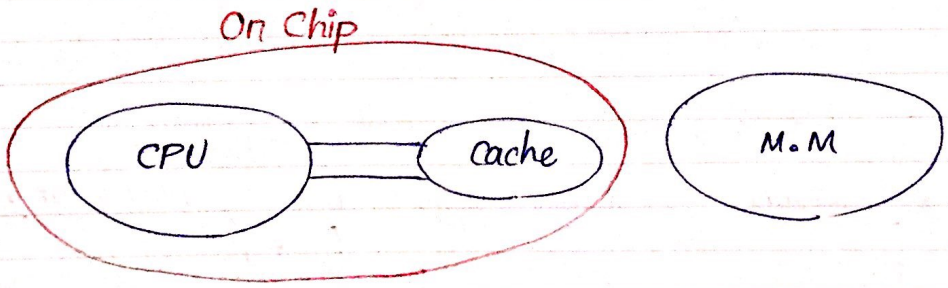
- Max اندازه Tag در دیتاپاه سوم است.

- توان مصرفی XOR ها بسیار زیاد است و در نتیجه گرمای بسیاری تولید می‌کند.

Access Time ← پاسخ‌گویی در چه زمانی است.

Lookup Time ← زمان جست‌وجو

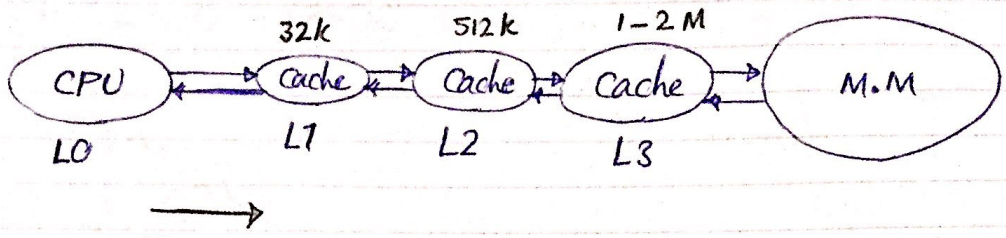
	hit Ratio	Access Time	Power	Area	Lookup Time
D.M.	−	Worst ++	−	−	Best −
K-W.A	+	+	+	+	+
F.A	++	Best −	++	++	Worst ++



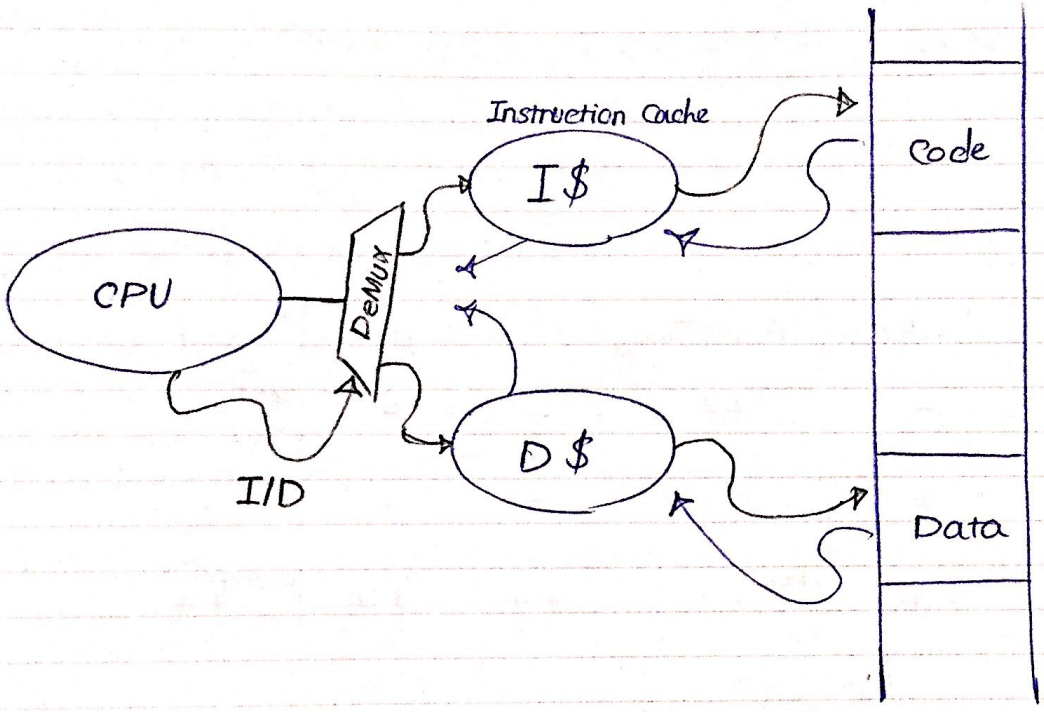
در مقابل onchip ← offchip

مثلاً onchip ← نمی‌توان CPU را بدون Cache استفاده کرد.

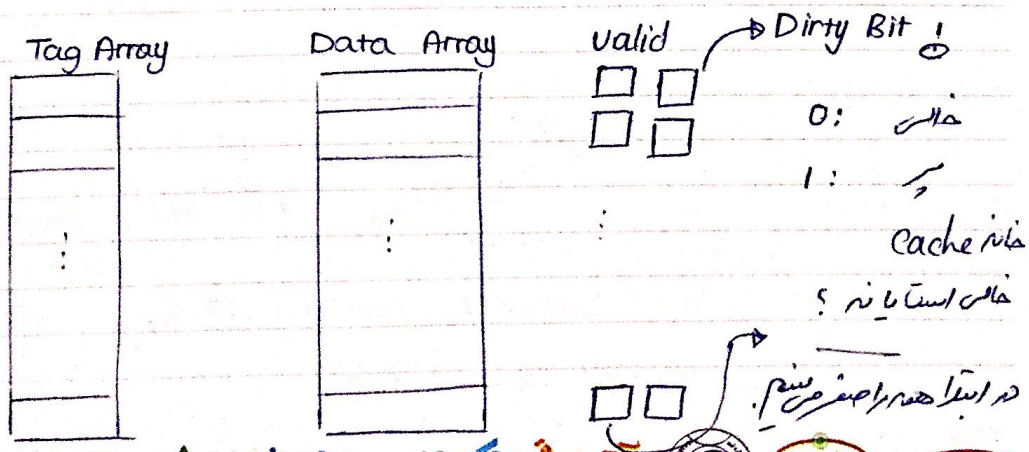
نمی‌توان Cache را تعویض کرد.



(برای تقسیم Cache به دو قسمت دیگر I. و دیگری D.
به hit ratio افزایش می‌یابد)



- تمام Cache ها روی Board (On Board) هستند.



Valid Bits فقط یک بار هنگام روشن شدن کامپیوتر مورد استفاده قرار می‌گیرند.

• انواع به روز رسانی:

1) Write Through (درون نویسی)

هنگام دستور write از طرف CPU، هم در cache مقدار را update می‌کنیم و هم در M.M.

در هر زمان، هر چیزی که در cache هست، در M.M هم هست.

این روش performance بدی دارد.

2) Write Back (پس نویسی)

اگر داده در cache از بین رود، در M.M لزوماً وجود ندارد.

در این روش هنگامی که می‌خواهیم داده را از cache حذف کنیم، داده را (آخرین مقدار) در M.M می‌نویسیم.

Dirty Bit به این منظور است که آیا این داده در M.M نوشته شده است یا خیر؟

سؤال: چنان چه اندازه حافظه اصلی 16 MB باشد و اندازه بلوک 64B باشد و اندازه حافظه نهان 64KB باشد، مطلوب است اندازه میدان‌ها در حالت:

1) B.M 2) 2_W.A

3) 4_W.A 4) F.A

1) $M = 2^4 * 2^{20} = 2^{24} \Rightarrow m = 24$

$B = 2^6 \Rightarrow b = 6$

به‌پخش فرامرفه آیچ‌تی یاکسان ایران مروارید سامون ساجنا گلتاش یاکسان

$C = 2^4 * 2^{10} \Rightarrow C = 14$

1)

23		b	0
8	10	6	

2) $k = 2$

23		b	0
9	9	6	

3) $k = 4$

23		0
10	8	6

4)

23	0
18	6

- Buffer فقط همجواری مکانی دارد ولی Cache همجواری مکانی و زمانی دارد.
- Cache قوی‌تر از Buffer است.

Arithmetic AND Logic Unit (ALU):

- نمایش اعداد:

۱- صحیح ←

A: بی‌علامت (Unsigned) $0 - (2^n - 1)$

B: علامت‌دار (Signed) به

- One's Complement
- Two's Complement
- Sign magnitude

Compute مشکل، I/O آسان
راحتی در Convert و تشخیص

Sign magnitude : [S] [n]

$-(2^n-1) \quad \dots \quad -0 \quad +0 \quad \dots \quad +(2^n-1)$

Two's Complement : $+(2^{n-1}-1) \quad \dots \quad -(2^{n-1})$

One's Complement : $+(2^{n-1}) \quad \dots \quad -(2^{n-1}-1)$

$\downarrow$

0 0

۲- اعشاری ←

A: ممیز ثابت

B: ممیز شناور

- در Two's Complement ← Compare مشکل است.

Convert مشکل است.

Range آن نصف می‌شود.

Compute آسان است. I/O مشکل است.

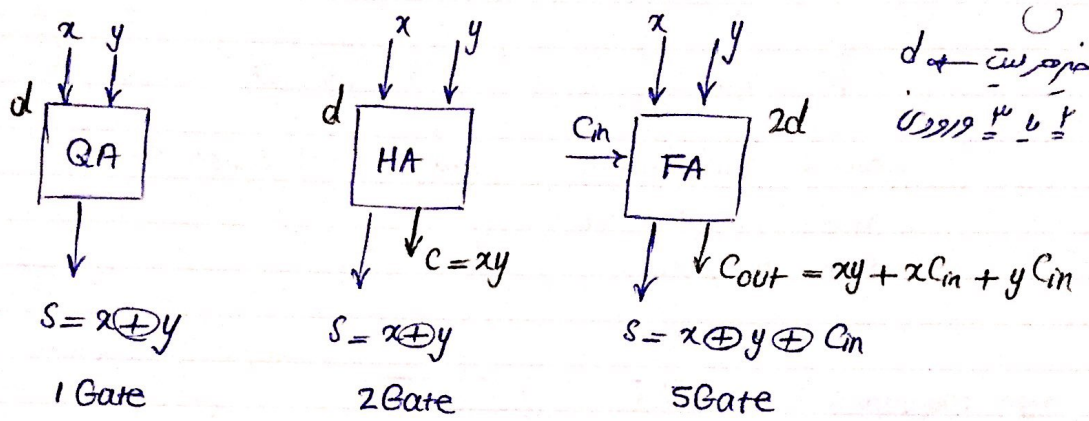
به‌پخش فراصدها آوا یاکسان ایران مروارید Salmon سایبا کتابشت یاکسان

- در One's C. : همه‌ی مشتقات + Two's دارای دو صفر است.

• جمع کننده :

۱) جمع کننده آبشاری

((Cascade (Ripple) Adder))

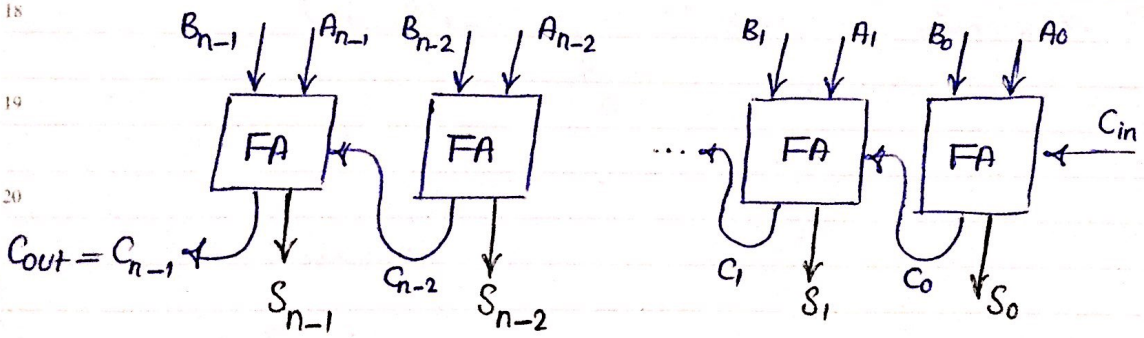


تاخیر هر بیت ← d

۲ یا ۳ ورودی

$\Delta = Delay_{FA} = 2d$

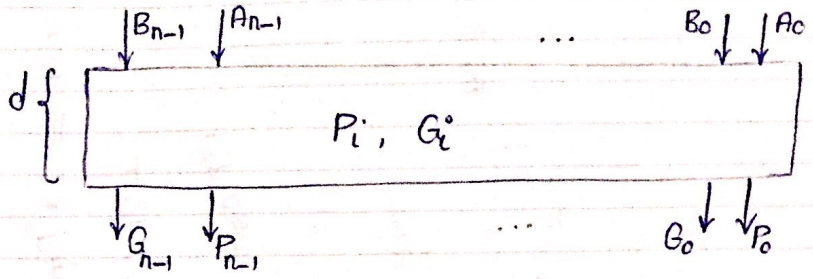
Ripple Adder :



هزینه : 5n Gate

$$\text{Delay} \begin{cases} C_{out} = 2nd \\ S = (2n-1)d \end{cases}$$

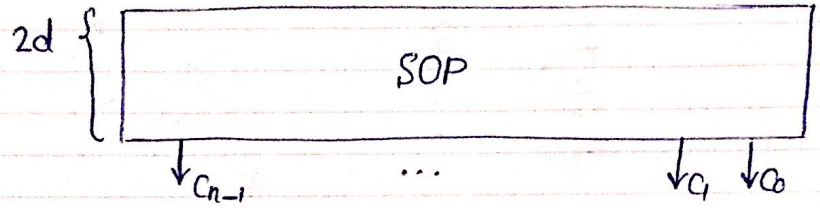
۲) جمع‌کننده پیش‌بینی‌گر رقم نقلی : « Carry Lookahead Adder »

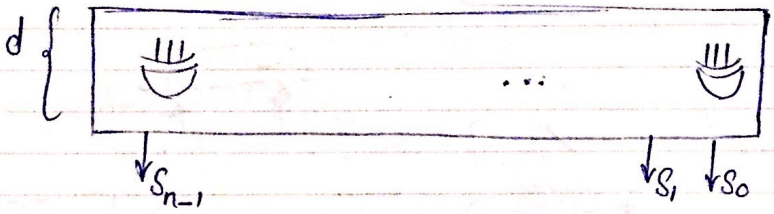


$P_i = A_i + B_i$

$G_i = A_i B_i$

. $2n$ Gate





n Gate

$$\Rightarrow \text{Delay} = \begin{cases} C_{out} = 3d \\ S = 4d \end{cases}$$

$\ldots n^2 + \ldots$ Gate

$= \frac{n^2 + 9n}{2}$

۳) جمع‌کننده انتخاب‌گر رقم نقلی « Carry Select Adder »

A : یکنواخت (Uniform)

B : غیر یکنواخت (Non-Uniform)

Uniform : A

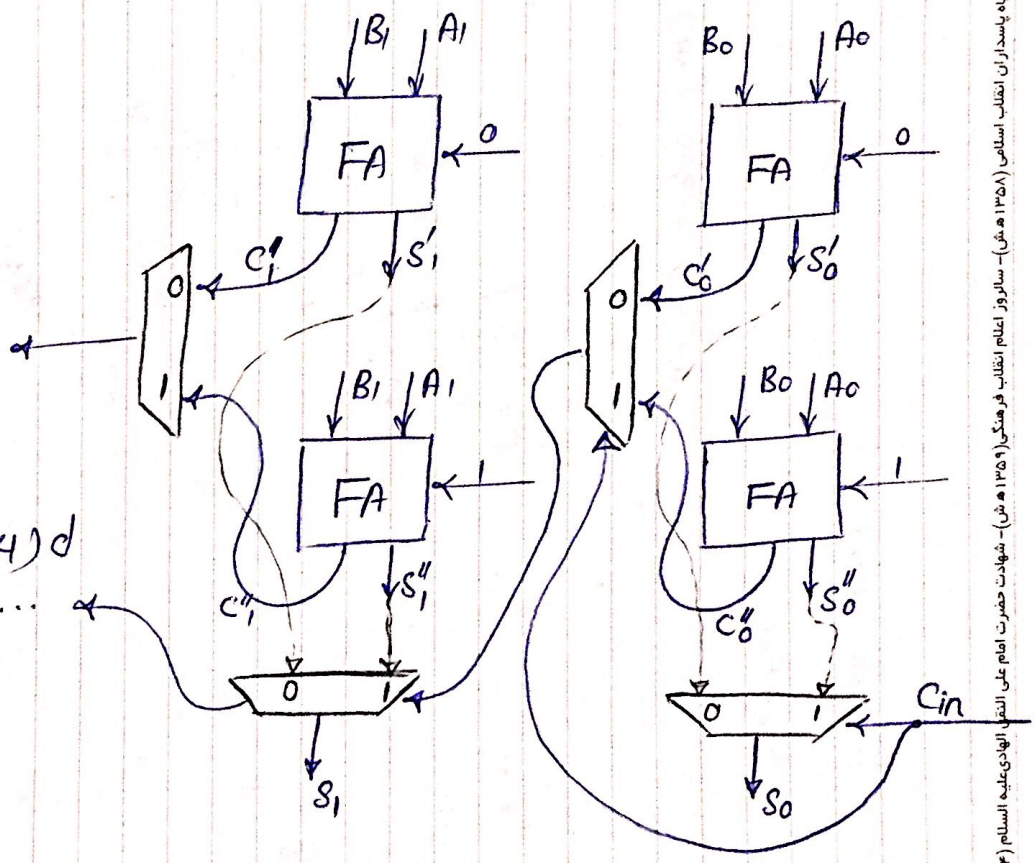
MUX → 3d , 4Gate

Cost = 2* Ripple Adder

+ 2n (4)

= 18n Gate

Delay
$$\begin{cases} Carry = (3(n-1)+4)\,d \\ S = (3(n-1)+4)\,d \end{cases}$$



تأسیس سپاه پاسداران انقلاب اسلامی (۱۳۵۸ ه ش)- سالروز اعلام انقلاب فرهنگی (۱۳۵۹ ه ش)- شهادت حضرت امام علی النقی الهادی علیه السلام (۲۵۴ ه ق)

B : Non_Uniform ←

۱ـ در مرحله اول لازم نیست با هر دو مقدار محاسبه کنیم. چون C_{in} در همان لحظه‌ی $t=0$ موجود است.

به جای این که یک مرحله را با مقادیر صفر و یک محاسبه کنیم، می‌توانیم، چندین مقدار را به ازای ورودی 0 را حساب کنیم.

$$1+1\overset{+1}{+}2\overset{+2}{+}4\overset{+1}{+}5\overset{+2}{+}7+\cdots+?=n$$

بعد از چند طبقه به n می‌رسیم؟ $L=?$

$L \simeq \sqrt{n}$ $\qquad$ $Delay = 3L-1$

$l \simeq \sqrt{3n}-1$

روز بزرگداشت شیخ بهایی

در طبقه اول یک بیت جمع می‌کنیم.

در طبقه دوم ۲ بیت جمع می‌کنیم.

در طبقه سوم ۴ بیت جمع می‌کنیم. و ...

$$Cost = (2n-1)\,FA + (L-1)(MUX_{2-1}) + (n-1)(MUX_{2-1})$$

$$= (2n-1)\,5 + 4(n+L-2)\ Gate$$

چنان طبقات با هم فرق دارند ← Non_Uniform

تأخیر از Ripple A. بهتر ولی از CLA بدتر است اما از آن جایی که نمی‌توان CLA را زیاد گسترش داد، CSA برای تعداد بالاتر مناسب‌تر است.

به‌پخش فراصفحه آی‌کیو یاکسان ایران مروارید سامون سایبان گلشاد یاکسان

Uniform Carry Select Adder (k bit) :

همانند ! بیتی است با این تفاوت که این بار در هر طبقه k بیت را با هم جمع می‌کنیم .

در هر طبقه می‌توان جمع‌کننده‌ها را CLA, RA و .. قرار داد . بدیهی است هرچه سرعت جمع‌کننده بیشتر باشد ، سرعت CSA بیشتر خواهد بود .

تعداد طبقات $= \lceil \frac{n}{k} \rceil$

Total Delay:

$S_{Delay} = (2k + 3L - 1)d \qquad L = \lceil \frac{n}{k} \rceil \qquad L \geq 2$

$C_{Delay} = (2k + 3L - 1)d \qquad L = \lceil \frac{n}{k} \rceil \qquad L \geq 2$

$\Rightarrow$ Total $Delay_{CSA} = (2k + 3\lceil \frac{n}{k} \rceil - 1)d$

$\Leftrightarrow (2k + 3\lceil \frac{n}{k} \rceil - 1)d < 2nd \qquad \exists k = ?$

این مسئله جواب دارد . بنابراین می‌توان k ای پیدا کرد که در رابطه بالا صدق کند .

$n = 8, 16, 32, 64$

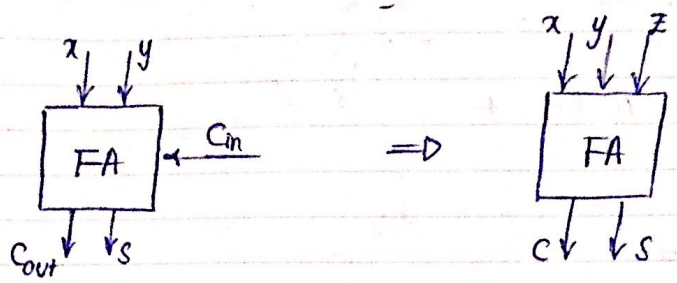
$n =$	8	16	32	64
k_{Min}				
k_{Best}				
k_{max}				

- چون تمامی طبقات یکسان هستند ، می‌توان محاسبات را تنها با تعداد محدودی طبقات مثلاً ۱ و ۲ و .. انجام داد .

به‌پخش فراصرفه یاکسان ایروان آرشه مروارید سامون سیاوش گلشاد یاکسان

" Carry Save Adder " (٤

حال می‌خواهیم m عدد n بیتی را با هم جمع کنیم.



تعداد فام‌ها برای این که دو عدد را با هم جمع بزنیم : $\log_{3/2}^{m}$

$$Cost_{L=1} = n * \left\lfloor \frac{m}{3} \right\rfloor$$

$$Cost_{L=2} = (n+1) * \left\lfloor \frac{m/(3/2)}{3} \right\rfloor \quad , \quad Cost_{L} = (n+\log_{3/2}^{m}) \left\lfloor \frac{m/(3/2)^{L}}{3} \right\rfloor$$

$$\Rightarrow {}^{*}Cost = \sum_{i=0}^{\log_{3/2}^{m}-1} \left((n+i) \frac{m/(3/2)^{i}}{3} \right)$$

$$Delay_{FullAdder} = \Delta = 2d$$

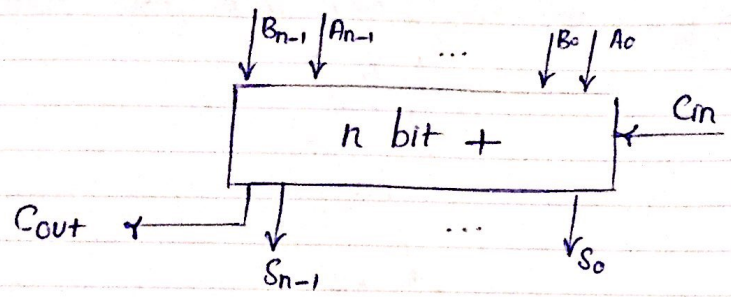
$$Delay = \log_{3/2}^{m} * \Delta + 2(n+\log_{3/2}^{m})d$$

$$\Rightarrow O(\tau) = (\log_{3/2}^{m} + n)$$

در مرحله آخر دو عدد را با RA با هم جمع می‌زنیم.

تعداد سطرها را در هر مرحله کم می‌کنیم ولی HA این گونه نیست (۲ سطر می‌برد ← ۲ سطر می‌دهد).

Overflow ← توسط یک بیت مشخص می‌شود.

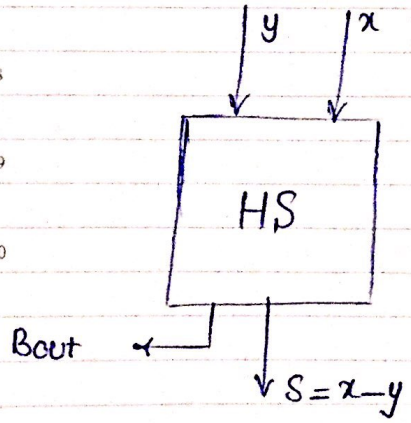


Example :

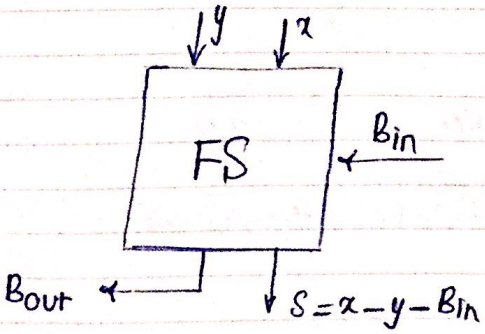
1	0000	0001
+ 255	+ 1111	1111
= 0	1 0000	0000

Carry Out = 1 ← Overflow در جمع بی علامت

• تفریق‌کننده‌ها : (Subtractors)



x	y	S = x − y	Borrow
0	0	0	0
0	1	1	1
1	0	1	0
1	1	0	0



x	y	B_{in}	$x-y-B_{in}$	B_{out}
0	0	0	0	0
0	0	1	1	1
0	1	0	1	1
0	1	1	0	1
1	0	0	1	0
1	0	1	0	0
1	1	0	0	0
1	1	1	1	1

$$\Rightarrow Sub = x \oplus y \oplus B_{in}$$

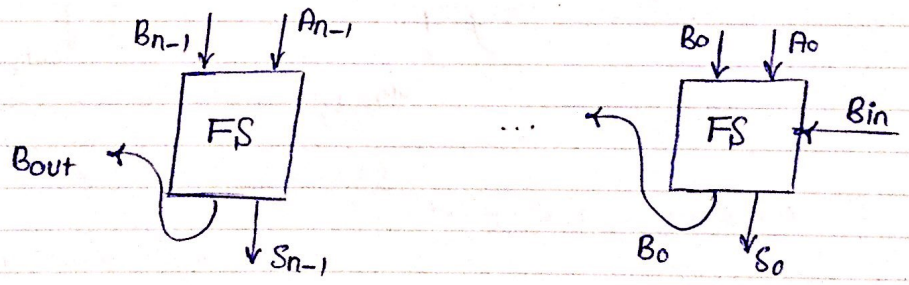
$$B_{out} = xyB_{in} + \bar{x}\bar{y}B_{in} + \bar{x}y\bar{B}_{in} + \bar{x}yB_{in}$$

$$= xyB_{in} + \bar{x}\bar{y}B_{in} + \bar{x}y$$

$$= \bar{x}y + yB_{in} + \bar{x}B_{in}$$

Delay : 3d Gates : 6_{Gate}

1) Ripple Subtractor :



هزینه : $n * FS = 6n_{Gates}$

Delay : $\begin{cases} Sub : (3n-2)d \\ B_{out} : 3nd \end{cases}$

$$\begin{array}{r} 12 \\ -\ 2 \\ \hline 10 \end{array} \qquad \begin{array}{r} 12 \\ -1\ \ 2 \\ -\ \ 6 \\ \hline -10\ \ 6 \end{array}$$

Borrow Out = 1 ← Overflow در تفریق بی علامت

2) $A+\bar{B}+1$

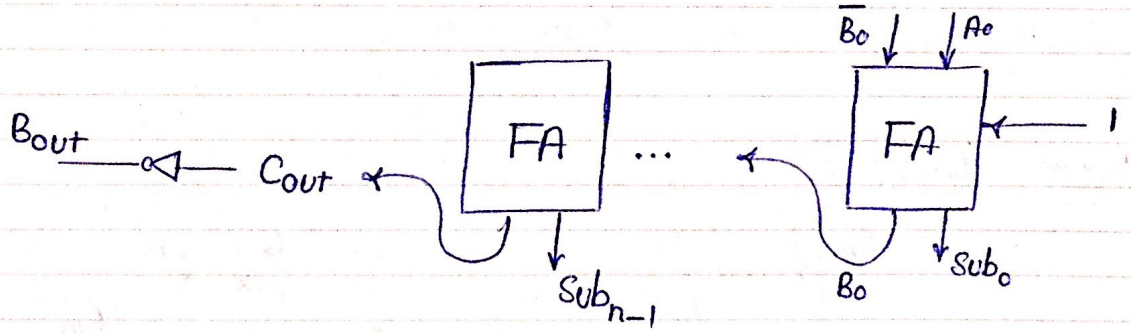
$A = A_{n-1} \cdots A_0$

$B = B_{n-1} \cdots B_0$

$$A-B = 2^n + A - B = A + (2^n - B) = A + \bar{B} + 1$$

n بیت

$$2^n - B = \bar{B} + 1$$



$A-B>0 \rightarrow C_{out}=1$

$A-B=0 \Rightarrow C_{out}=1$

$A-B<0 \rightarrow C_{out}=0$

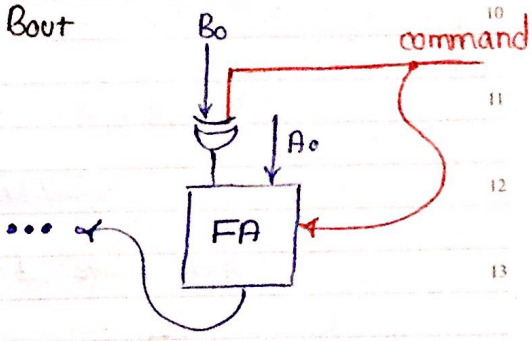
Overflow $\rightarrow C_{out}=0$

به پخش فراصرفه پاکسان ایران مروارید سامون سایپا گلشاد پاکسیان

به جای FA هر Adder دلخواهی می‌توان قرار داد.

هزینه: $6nGate$

Delay: $(2n+2)d \rightarrow B_{out}$



$$\text{Command} \begin{cases} 0 \rightarrow \text{Add} \\ 1 \rightarrow \text{Subtract} \end{cases}$$

مزیت‌ها ← ۱- سرعت بهتر

۲- می‌توان با سخت‌افزار جمع share شود.

روز ملی خلیج فارس

آیا می‌شود Borrow Lookahead ساخت؟

آیا می‌شود Borrow Select ساخت؟

• نمایش ململ ۲ (علامت‌دار)

$A = (0 \ldots)_2$

$-A = \bar{A}+1 = (1 \ldots)_2$

روش $A+\bar{B}+1$ برای ململ ۲ درست است ولی روش اول (Cascade Subtractor) برای ململ ۲ نادرست است (مثال نقض)

روز جهانی کار و کارگر

به‌پخش فرامرفه یاکسان ایروان مروارید Nonim ساجنا کلکاشت یاکسان

در مکمل ۲

$$\begin{array}{rl} & 1000\ 0000 \quad (-128) \\ + & 1000\ 0000 \quad (-128) \\ \hline & 1\ \underbrace{0000\ 0000}_{=0} \end{array}$$

Cout

در n بیت

← Adder درست کار نمی‌کند.

☹ مواقعی که جواب جمع‌کننده‌ها برای مکمل ۲ غلط است؟

علامت جواب ≠ علامت A, B

if $A_{n-1} B_{n-1} \overline{S_{n-1}} + \overline{A_{n-1}}\,\overline{B_{n-1}} S_{n-1} = 1 \rightarrow$ False (Wrong Answer)

$++ = -$ ← Overflow

$-- = +$

if $A_{n-1} B_{n-1} \overline{S_{n-1}} + \overline{A_{n-1}}\,\overline{B_{n-1}} S_{n-1} = 1$ ①

$\equiv$ if $C_{n-1} \oplus C_{n-2} = 1$

Overflow برای تفریق مکمل ۲ ←

$+ - = -$

$- + = +$

if $\overline{A_{n-1}} B_{n-1} S_{n-1} + A_{n-1} \overline{B_{n-1}}\,\overline{S_{n-1}} = 1$

Cout ← در Carry Flag ذخیره می‌شود. (در بی علامت)

Overflow ← در Overflow Flag ذخیره می‌شود. ← عبارت ①

به‌نشر فراصفحه یاکسان ایران مروارید سامون ساجنا کتاباش یاکسان

۵) چنان چه دو عدد ۸ بیتی مد نظر باشد و عملیات جمع و تفریق $(A+\bar{B}+1)$ داشته باشیم:
مثالی از دو عدد بنویسید:

۱) در منطق بی علامت، پاسخ درست باشد؟

۲) در منطق بی علامت، پاسخ نادرست باشد. (Overflow)

۳) در مکمل ۲ پاسخ درست باشد.

۴) در مکمل ۲ پاسخ نادرست باشد.

۵) اگر بی علامت نگاه شود، جواب صحیح و اگر مکمل ۲ نگاه شود، جواب نادرست است.

۶) برعکس اتفاق (۵) بیفتد.

1)
```
  0000 1000        8
+ 0001 0011      + 19
----------       ----
  0001 1011        27 ✓
```

2)
```
  0001 0000        16
+ 1111 1111      + 255
----------       -----
1 0000 1111 →     15 ✗
```

3)
```
  0000 0110        6
+ 0000 1000      + 8
----------       ---
  0000 1110        14 ✓
```

4)
```
  0110 0000        +96
+ 0110 0000        +96
----------       ----
  1100 0000        -64 ✗
```

5)

```
  0000 0001
+ 0111 1111
-----------
  1000 0000
```

مکمل ۲ : +1
+ +127
−128 ✗

بی علامت : +1
+ +127
+128 ✓

6)

```
  1000 0001
+ 0111 1111
-----------
 10000 0000
```

مکمل ۲ : −127
+ +127
0 ✓

بی علامت : 129
+ +127
256 ✗

؟ مشخص کنید وضعیت ثباتهای پرچم للام مزینه است؟ (چنان چه توازن زوج باشد).

A = 0110 0000
B = 1010 1110

روش : $A+\bar{B}+1$

C	O	Z	P
1	0	1	0

۱)

1	0	0	0

۲)

0	1	1	1

۳)

0	1	1	1

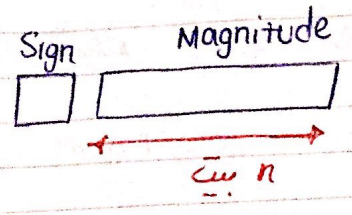
۴)

0	1	0	0

```
  0110 0000
+ 0101 0010
-----------
0 1011 0010
```


به نشر
هراصرفه
یاکسان ایروان
مروارید
سلمون
ساحل
کتاب
یاکسان

روش اندازه علامت :



اگر ثابت بنید چنانچه دو عدد بی علامت X , Y بخواهند مقایسه شوند می توان $X+\bar{Y}+1$

$X \geq Y \longleftrightarrow C=1$ اگر : A

$X < Y \longleftrightarrow C=0$ اگر : B

$S_A|A| + S_B|B| \longrightarrow S_A \oplus S_B \xrightarrow{1}$ مختلف العلامت

$\xrightarrow{0}$ هم علامت

• هم علامت $\longrightarrow |A|+|B| \rightarrow |C|$

AVF $\rightarrow$ Overflow

$S_C \leftarrow S_A$

$E|C| = |A|+|B|$

$AVF \leftarrow E$

• مختلف العلامت $\longrightarrow EC \leftarrow A+\bar{B}+1$ $\quad S_C \leftarrow S_A$

if (C=0) $S_C \leftarrow 0$

if (E=1) $\rightarrow A>B \rightarrow S_C \leftarrow S_A$

if (E=0) $\rightarrow A<B \rightarrow C \leftarrow \bar{C}+1$ Borrow=1 ملحل ۲ Borrow=0

$S_C \leftarrow \overline{S_C}$

$\longrightarrow$ Answer : S_C C

if (AVF=1) Wrong Answer

روز بزرگداشت شیخ صدوق

مشکل ← تولید دو صفر $(+0, -0)$

هم در Delay و هم در Area ضعیف‌تر است (در جمع)

اما در ضرب و تفریق برتری است.

• تفریق : $S_A A - S_B B \longrightarrow S_A A + \overline{S_B} B$

همانند الگوریتم فوق

با : $S_A \oplus S_B \longrightarrow$ مختلف العلامة : 0

هم علامت : 1

به‌پخش فرامرفه آرکی‌فی یاگدیان ایران مروارید سلمون سیاجنا کتاباش یاگدیان

۱۳۹۴
پنجشنبه
۱۷
اردیبهشت
THURSDAY
7 May 2015
۱۸ رجب ۱۴۳۶

08
09
10
11
12
13
14
15

۱۳۹۴
جمعه
۱۸
اردیبهشت
FRIDAY
8 May 2015
۱۹ رجب ۱۴۳۶

روز بیماریهای خاص و صعب العلاج

به‌پخش فراصرفه آتیه یاکسان ایروان مروارید هامون ساجنا گلشاد یاکسان

۱۳۹۴
شنبه
۱۹
اردیبهشت
SATURDAY
9 May 2015
۲۰ رجب ۱۴۳۶

08
09
10
11
12
13
14
15
16
17
18
19
20

روز بزرگداشت شیخ کلینی

به‌پخش
فراصرفه
آی‌کف
یاکسان ایروان
مروارید سامون
ساجنا
گلتاش
یاکسان

تمرین

• تمرین: ۱) ثابت کنید برای اعداد ململ ۲ داریم:

$$x_{n-1}\, x_{n-2} \dots x_2\, x_1\, x_0$$

$$= (x_{n-1})\,(-1)^{x_{n-1}} \times 2^{n-1} + \sum_{i=0}^{n-2} x_i 2^i$$

۲) مدار ضرب بسته آرایه‌ای را برای اعداد ململ ۲، هشت بیتی ترسیم کنید و هزینه و تأخیر آن را بیابید.

• در هنگام افزایش بیتی از فضای کوچک‌تر به بزرگ‌تر ←

- بی علامت ← صفر بیت علامت گذاشته می‌شود (zero padding)

- ململ ۲ و ململ ۱ ← علامت (چپ‌ترین بیت) تکرار می‌شود (Sign Extension)

Shift:

- SHR (Shift Right)
- SHL (Shift Left)
- SAR (Shift Arithmetic Right) شیفت حسابی به راست
- SAL (Shift Arithmetic Left) = SHL

SHR → ÷2 بی علامت ←

SHL → x2

01101010 $\xrightarrow{SHR}$ 00110101

$\xrightarrow{SHL}$ 11010100

یاکسان کتابشت ساجد مروارید سلمون یاکسان ایران آشنا فراصرفه به نخش

SAR → 00110101

SAL → SHL همان

10001010 SHR → 01000101

SHL → 00010100

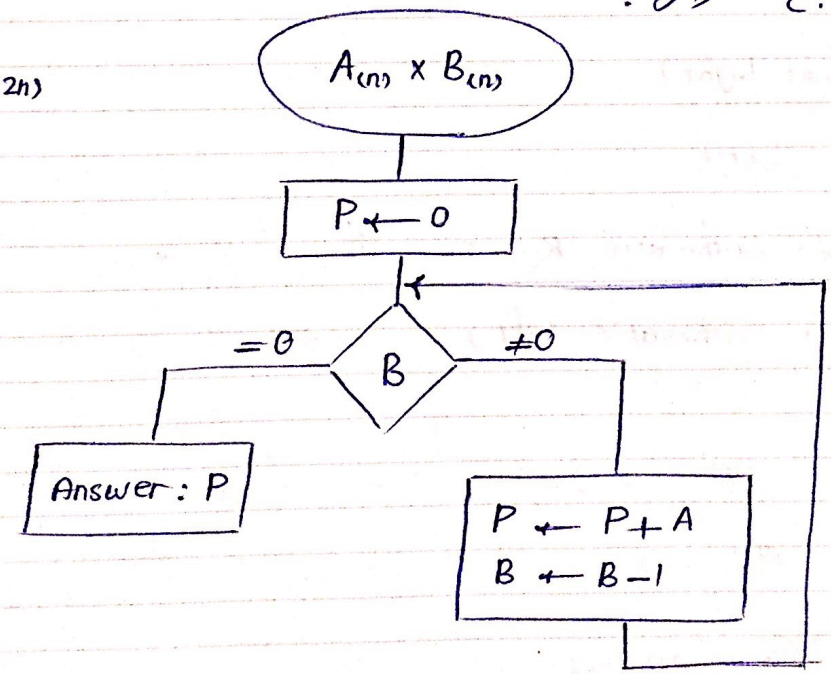
SAR → 11000101

SAL → SHL همان

SAL	SAR	SHL	SHR	
×2	×	×2	÷2	بی علامت
×	÷2	×	×	علامت دار

۲ـ ضرب توسط جمع‌های متوالی :

$P_{(2n)}$



به‌نشر فرامرفه آی‌تی‌کی ایران یاکسان مروارید سامون ساجنا کتاشت یاکسان

۳- ضرب به وسیله‌ی شیفت و جمع :

$$\begin{array}{lll} & A_{n-1}\ A_{n-2} \cdots A_2\ A_1\ A_0 & \\ \times & B_{n-1}\ B_{n-2} \cdots B_2\ B_1\ B_0 & \\ \hline + & \begin{matrix} 0 & & 0 \\ & & \\ A & & 1 \end{matrix} \Big\} B_i & i\ SHL \end{array}$$

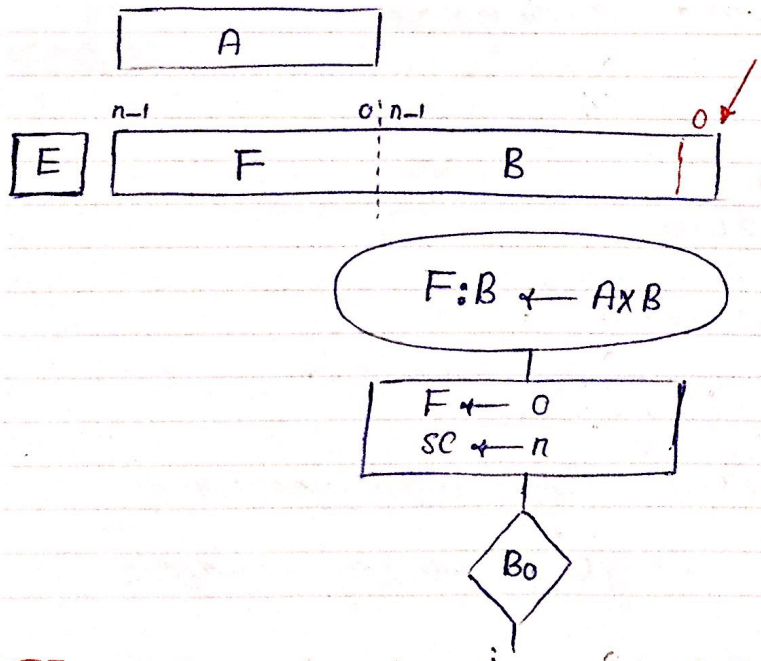
Register ها 2n بیتی خواهند بود .

Product (2n Bit)

n بیت کم ارزش مشخص می‌شوند ← n بیت بعدی در مرحله آخر

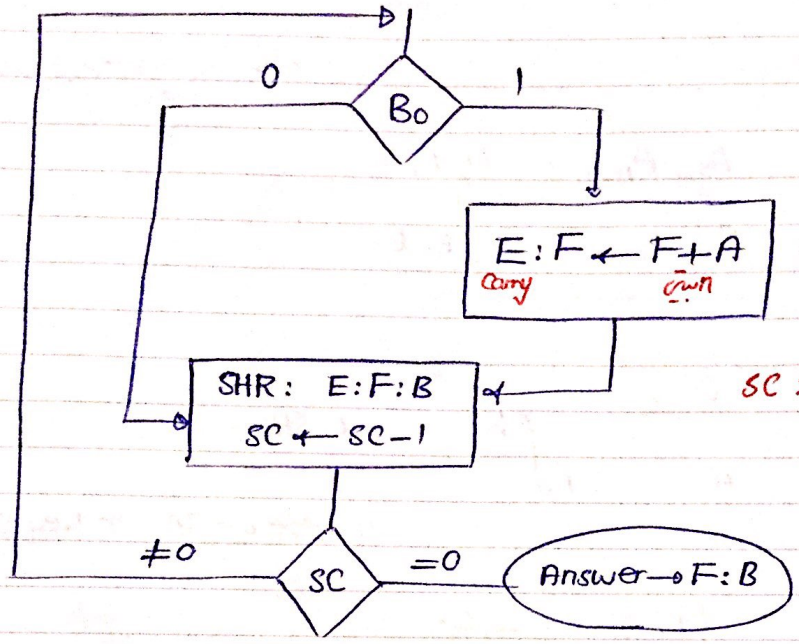
به جای استفاده از شمارنده ← مضروب فیه را به سمت راست شیفت می‌دهیم .

به جای این که Product های میانی را به سمت چپ شیفت دهیم ، Product پایانی را در هر مرحله به سمت راست شیفت می‌دهیم .



اما زمان و سخت افزار کم‌تری نسبت به ۲ دارد .

Max Add = n



SC : Sequence Counter

برای من علامت درست است ولی برای مکمل ۲ کار نمی‌کند.

اگر B منفی باشد ← مکمل ۲ ← شیفت و جمع ← مکمل ۲

٤: ضرب بیتی بوث (Booth Multiplier):

مکانیزم بوث ← در مقداری از بیت‌ها ← اولین بیت : تبدیل به (−1) می‌شود.

بیت بعد از آخرین ۱ : تبدیل به (+1) می‌شود.

01100111

(+1) 0(0)0(+1) 00 (−1)

$+2^7 - 2^5 + 2^3 - 1 = 103$

1110 1111

(+1) 00 (−1) (+1) 000 (−1)

اگر ۱+ آخری در نظر گرفته شود ← من علامت

در غیر این صورت ← با علامت (مکمل ۲)

$+2^8 - 2^5 + 2^4 - 1 = 239$

مثال: 0 1 0 1 1 0 1 1

(+1) (−1)(+1) 0(−1)(+1) 0 0(−1)

↓

بی علامت = 183

در مکمل ۲ نمی‌نویسیم

مکمل ۲ = −73

اگر چپ‌ترین بیت، صفر باشد ← بی علامت = مکمل ۲

1111 0000

(−1) → مکمل ۲ = −16

(+1) (−1) → بی علامت = 240

0 1 0 1 0 1 0 1

(+1)(−1) (+1) (−1) (+1) (−1) (+1) (−1)

$\rightarrow 2^7 - 2^6 + 2^5 - 2^4 + 2^3 - 2^2 + 2 - 1$

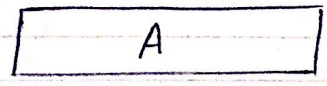
لغو امتیاز تنباکو به فتوای آیت الله میرزا حسن شیرازی (۱۲۷۰ هـ ش) - شهادت حضرت امام موسی کاظم(ع)(۱۸۳ هـ ق)

1 0 1 0 1 0 1 0

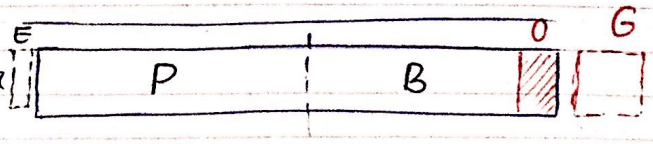
(−1) (+1) (−1) (+1) (−1) (+1) (−1) → مکمل ۲

(+1) (−1) (+1) (−1) (+1) (−1) (+1) (−1) → بی علامت

n بیت



2n بیت



روز بزرگداشت فردوسی

به‌پخش فراصرفه آرشه یاکدیان ایران مروارید سامون ساجنا کتاباش یاکدیان

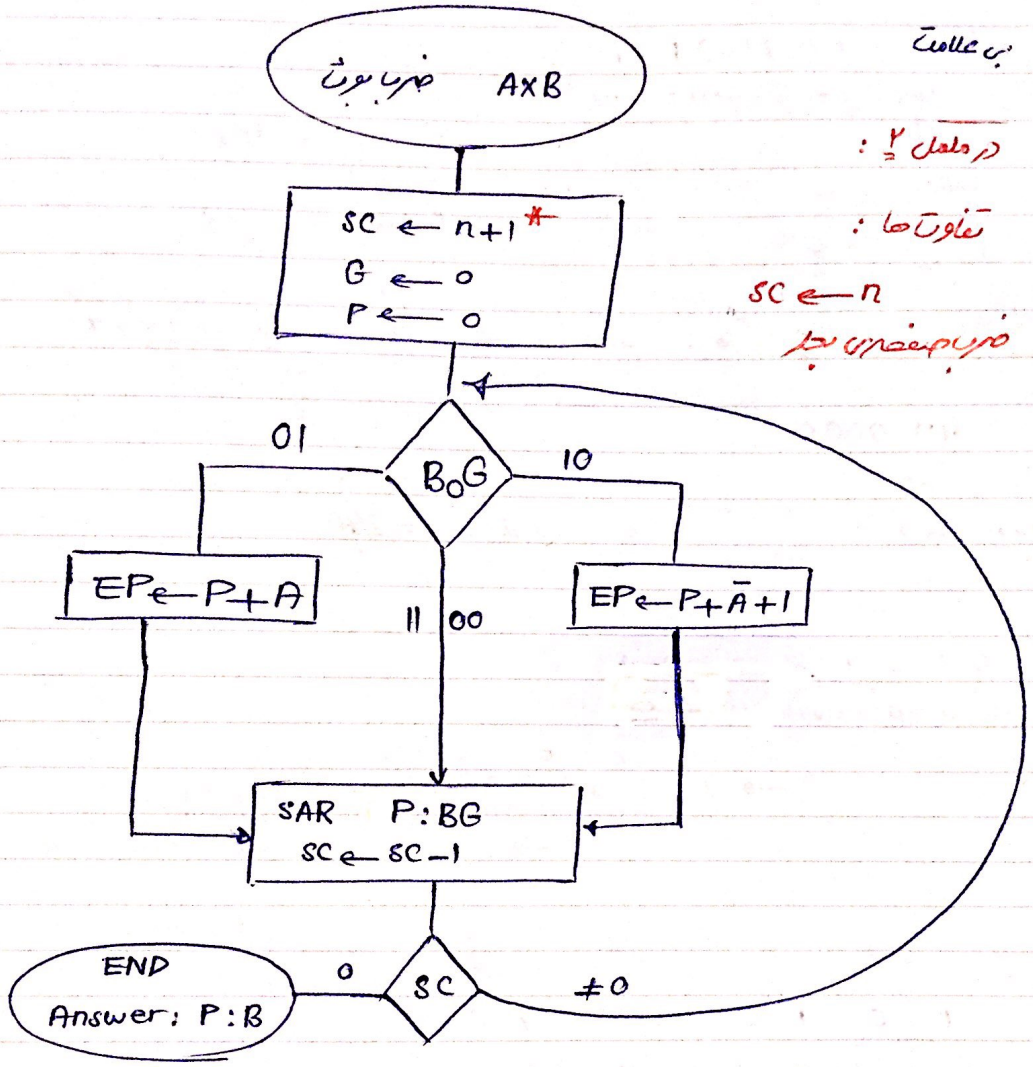
بی علامت

در حالت ۲ :

تفاوت ها :

$SC \leftarrow n$

ضرب صفحه ای بعد



مکمل ۲ :

```
        0 1 1 0        6
      X 1 0 1 0       -6
      (-1) (+1) (-1)
  ---------------------
  0 0 0 0 0 0 0
  1 1 1 0 1 0 0
  0 0 1 1 0 0 0
  1 0 1 0 0 0 0
  ---------------------
  1 1 0 1 1 1 0 0      = -36
```

بی علامت :

```
        0 0 1 1 0      6
      X 0 1 0 1 0      10
      (+1) (-1) (+1) (-1)
  ---------------------
  0 0 0 0 0 0 0 0 0
  1 1 1 1 1 0 1 0 0
  0 0 0 0 1 1 0 0 0
  1 1 1 0 1 0 0 0 0
  0 0 1 1 0 0 0 0 0
  ---------------------
  1 0 0 0 1 1 1 1 0 0  = 60
```

تقسیم شده :

$A_{(n)} \times B_{(n)} \longrightarrow P_{(2n)}$

$\frac{(2n)}{\text{مقسوم}} \div \frac{(n)^B}{\text{مقسوم علیه}} \longrightarrow$ خارج قسمت $(n) \xrightarrow{P} A_2$ ، باقی مانده $(n) \xrightarrow{P} A_1$

$A_1(n) : A_2(n)$

نمی توان تضمین کرد که خارج قسمت در n بیت جا شود.

هرگاه خارج قسمت در n بیت جا نشود ← overflow رخ می دهد.

به طور مثال ← $\div 0 = \infty$

شرط لازم و کافی :

n بیت سمت چپ مقسوم $\geq$ n بیت مقسوم علیه $\longrightarrow$ overflow

$A_2 \; A_1 \; — \; | \; B_2 \; B_1$

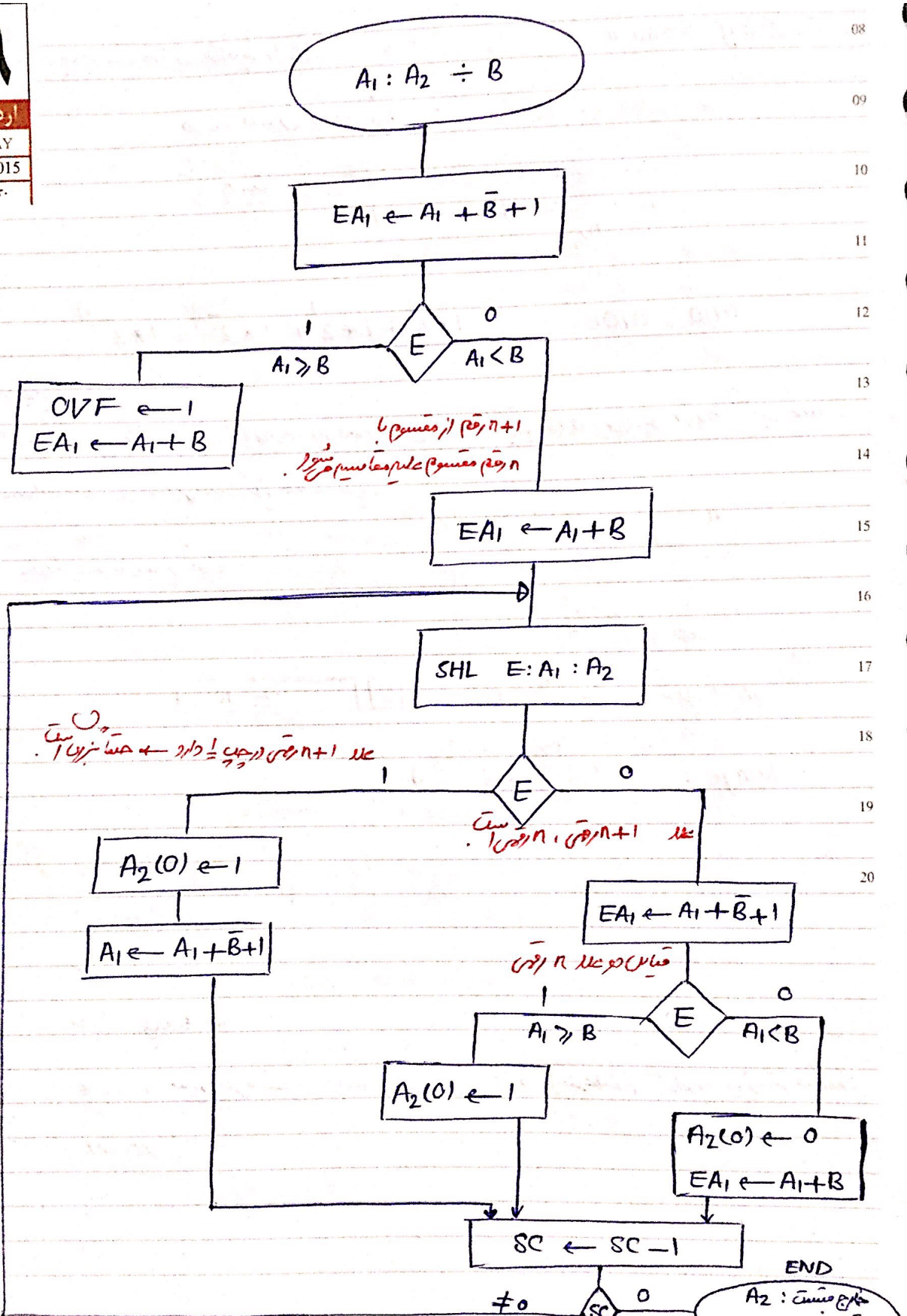
$A_2 : A_1 \geq B_2 : B_1 \longrightarrow$ overflow

overflow $\begin{cases} \div 0 = \infty \\ A_1 \geq B \end{cases}$ دو حالت، یکسان می باشند $\longrightarrow A_1 \leq B$

اگر x ، y دو عدد n بیتی باشند :

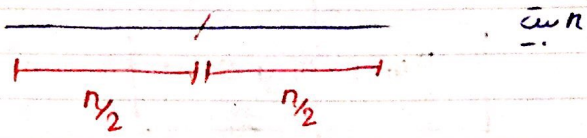
$x + \bar{y} + 1 \longrightarrow \begin{cases} \text{Carry} = 1 \longrightarrow x \geq y \\ \text{Carry} = 0 \longrightarrow x < y \end{cases}$

به پخش فراصرفه پاکسیان ایروان مروارید سامون ساجنا کتاباش پاکسیان


A1 : A2 ÷ B
EA1 ← A1 + B̄ + 1
E
1
A1 ≥ B
0
A1 < B
OVF ← 1
EA1 ← A1 + B
n+1 رقم از مقسوم با
n رقم مقسوم علیه مقایسه می‌شود.
EA1 ← A1 + B
SHL E: A1 : A2
علامت n+1 رقمی در چپ ۱ دارد ← حتماً بزرگتر است.
E
1
0
علامت n+1 رقمی، n رقمی است.
A2(0) ← 1
A1 ← A1 + B̄ + 1
EA1 ← A1 + B̄ + 1
مقایسه دو علامت n رقمی
E
1
A1 ≥ B
0
A1 < B
A2(0) ← 1
A2(0) ← 0
EA1 ← A1 + B
SC ← SC − 1
SC
≠ 0
0
END
خارج قسمت : A2
باقی مانده : A1

ممیز شناور در نمایش اعداد : "Floating Point"

در حالت معمول (ممیز ثابت) :



$0110 / 0101$ $\qquad 1*2^{2}+1*2^{1}+1*2^{-2}+1*2^{-4}$

در ممیز ثابت :

برای جمع و تفریق با قرار دادن ممیزها زیر هم، می‌توان از جمع کننده‌ها و تفریق کننده‌های بی علامت استفاده کرد. برای ضرب نیز هم چنین.

حداکثر عدد در ممیز ثابت : $2^{n/2}-1 \;/\; 1-2^{-n/2}$

دقت : $2^{-n/2}$

تعداد اعداد : $2*2^{n}-1$ $\qquad$ [S] [n بیت]

Range : $\pm(2^{n/2}-2^{-n/2})$

مشکلات ممیز ثابت :

۱ـ دقت کم

۲ـ وجود خطا

۳ـ Range کم

۴ـ کاربرد بیش‌تر قسمت صحیح در برخی استفاده‌ها و یا برعکس، کاربرد بیش‌تر قسمت اعشاری

به‌پخش فرامرفه آیکف یاگسیان ایروان مروارید سامون ساجنا گلناش یاگسیان

نمایش Floating Point بر اساس استاندارد IEEE 754 :

اعداد هنجار (Normal) ←

$+4.32 * 10^{-3}$

$-1.2432 * 10^{6}$

اعداد ناهنجار (UnNormal) ←

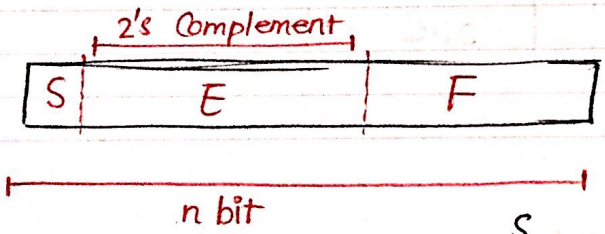
$+10$ $\xrightarrow{\text{Normal}}$ $+1.0 * 10^{+1}$

$+253,2$ $\xrightarrow{\text{Normal}}$ $+2.532 * 10^{+2}$

-0.4530 $\xrightarrow{\text{Normal}}$ $-4.53 * 10^{-1}$

110101.11 $\xrightarrow{\text{Normal}}$ $1.1010111 * 2^{+5}$

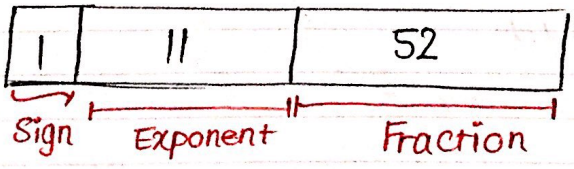
Binary $\xrightarrow{\text{Normal}}$ $\pm 1.F * 2^{E}$



$= (-1)^{S} * 1.F * 2^{E}$

$n = 32$ (Single Precision)

$n = 64$ (Double Precision)



به نشر فراصرفه یاکسان ایروان مروارید سلمون ساجنا گلناشت یاکسان

S	E	F
1	3	4

$+5.6$

$+101.10011001\ldots$

$= +1.0110011001\ldots * 2^{+2}$

0	010	0110

-0.711

$-0.1011011\ldots$

$= -1.011011 * 2^{-1}$

1	111	0110

$+24.51$

$+11000.10000\ldots$

$= +1.10001000\ldots * 2^{+4}$

$+3 \ldots -4$

0		1000

$\longrightarrow$ overflow

-0.03830

-0.0000100111

$= 1.00111\ldots * 2^{-5}$

$\longrightarrow$ overflow

به نخش فراصرفه آرشیو پاکسان ایران مروارید سامون ساجنا کتاب پاکسان

$+3.75$

$+\ 11.11$

$= +1.11100\cdots * 2^{+1}$

0	001	1110

-0.25

$-0.01 = -1.00\cdots * 2^{-2}$

1	110	0000

- صفر در نمایش هنجار قابل نمایش نیست.

مثال: نمایش ممیز شناور ←

۱- بزرگترین عدد مثبت؟

۲- کوچکترین عدد مثبت؟

۳- تعداد اعداد قابل نمایش؟

۴- دقت نمایش؟

$(-1)^S\ 1.F * 2^E \equiv$

1	e	f
S	Exponent	Fraction

$$\begin{cases} F_{min} = .000\ldots0 = 0 \\ F_{max} = .111\cdots1 = 1-2^{-f} \end{cases}$$

به‌پخش فراصرفه یاکسان ایروان سامون موبایل ساجنا کتاباش یاکسیان

$$\begin{cases} E_{min} = 100\ldots0 = -2^{e-1} \\ E_{max} = 011\ldots1 = 2^{e-1}-1 \end{cases}$$

$$N_{Max} = 1.F_{max} * 2^{E_{max}} = (2-2^{-f}) * 2^{+(2^{e-1}-1)}$$

$$\varepsilon = 1.F_{min} * 2^{E_{min}} = 2^{-2^{e-1}}$$

بازه نمایش $\longrightarrow [-N_{max}, N_{max}]$ کل بازه نمی‌باشد ← چون فقط n بیت در اختیار داریم

- در Floating Point اصطلاح overflow بی‌معنی است (به جز $\div 0$) چون بی‌شمار عدد حقیقی وجود دارد.

ε : موجب‌ترین عدد مثبت $= 1.F_{Min} * 2^{E_{Min}} = 1.00\ldots0 * 2^{E_{Min}} = 2^{-2^{e-1}}$ ①

دومین عدد مثبت $= 1.000\ldots01 * 2^{E_{Min}} = (1+2^{-f})\,2^{-2^{e-1}}$ ②

سومین عدد مثبت $= 1.00\ldots010 * 2^{E_{Min}} = (1+2^{-f}+2^{-f})\,2^{-2^{e-1}}$

$\Delta = 2^{-f} * 2^{E_{Min}}$

⋮

امین عدد مثبت $2^f = 1.111\ldots11 * 2^{E_{Min}} = (2-2^{-f})\,2^{-2^{e-1}}$ $\rightarrow \Delta$

اشکو ۱

$1.00\ldots0 * 2^{E_{min}+1} = 2^{-2^{e-1}+1} = 2\varepsilon$

$1.00\ldots01 * 2^{E_{min}+1} = (1+2^{-f})\,2^{-2^{e-1}+1}$

$\Delta' = 2^{-f} * 2^{E_{Min}+1} = 2\Delta$

اشکو ۲

⋮

$1.111\ldots1 * 2^{E_{min}+1} = (2-2^{-f})\,2^{-2^{e-1}+1}$ $\rightarrow \Delta'$

⋮

به‌پخش فراصرفه پاکسان ایران مروارید سامون سایپا کشاورز پاکسان

$$1.111\ldots1 * 2^{E_{max}} = (2-2^{-f}) 2^{2^{e-1}-1}$$

2^e اشتو

$$② - ① = \Delta = 2^{-f} * 2^{E_{min}}$$

$$\Delta_3 = 2^{-f} * 2^{E_{min}+2} = 2\Delta_2 = 4\Delta_1$$

$$\Rightarrow \Delta_{2^e} = 2^{2^e-1}\Delta_1$$

$$\Delta_1 = 2^{-f} * 2^{E_{min}}$$ ← دقت نمایش :

$$E_{min} = -2^{e-1}$$

فاصله در اشتوهای بالا بسیار زیاد است ← جمع عدد بسیار بزرگ با عدد بسیار کوچک قابل نمایش نیست.

$$S_1 = \underbrace{1 + 1/2 + 1/3 + \cdots + 1/n}_{\text{بزرگ}} \quad \text{کوچک}$$

$$S_2 = \underbrace{1/n + 1/(n-1) + \cdots + 1/2}_{\text{بزرگ}} + \underbrace{1}_{\text{بزرگ}}$$

S_2 دقیق‌تر است.

1	4	3
S	E	F

$Biased_1 = 2^e - 1$

$E_{Min} = -2^{e-1} = 2's\ Complement + Biased$

$= \underbrace{00\ldots0}_{e}$

+8 +7

Exponent	ملحل ۲ =	$Biased_1$	$Biased_2$
+7	0111	1111	1110
+6	0110	1110	1101
+5	0101	1101	1100
+4	0100	1100	1011
+3	0011	1011	1010
+2	0010	1010	1001
+1	0001	1001	1000
0	0000	1000	0111
-1	1111	⋮	0110
-2	1110		⋮
⋮			
-7	1001	0001	0000
-8	1000	0000	1111

بی‌علامت

بجز آخر بی‌علامت

۱۳۹۴
پنجشنبه
۷
خرداد
THURSDAY
28 May 2015
۹ شعبان ۱۴۳۶

if $E = 00 \ldots 0$

AND $F = 00 \ldots 0$ THEN Number $= 0$

AND $S = 0$

OR $S = 1$

اعدادی مانند π , e را نیز مانند صفر وارد این نوع نمایش می‌کنیم.

NaN $\rightarrow$ Not a Number : $\pi, e, +\infty, -\infty, \sqrt{2}$

$\text{Biased}_2 \rightarrow$ if $E = 11 \ldots 11 \rightarrow$ NaN

۱۳۹۴
جمعه
۸
خرداد
FRIDAY
29 May 2015
۱۰ شعبان ۱۴۳۶

به‌نشر فرامرفه آی‌تی‌کده یاکسان ایروان مروارید Nonim سایبان کتاشت یاکسان

0	0000	000

صفر ←

• جمع دو عدد ممیز شناور : (پیچیده تر از ممیز ثابت)

S_A	E_A	F_A

$(-1)^{S_A} \, 1.F_A \times 2^{E_A - b}$

S_B	E_B	F_B

$(-1)^{S_B} \, 1.F_B \times 2^{E_B - b}$

$\overset{+}{D}$?

توان ها را برابر می کنیم ← سپس با هم جمع می کنیم .

برای برابر کردن توان ها ← ۱ : افزایش توان ← SHR

۲ : کاهش توان ← SHL

اگر عدد با نمای کوچک تر را به عدد با نمای بزرگ تر برسانیم ، بهتر است ؛ چون اگر عددی هم از بین برود ، ارزش کم تری دارد .

الگوریتم جمع ← (و تفریق)

۱) چک کردن صفر

۲) نماها هم ردیف شود (عدد با نمای کوچک به عدد با نمای بزرگ برسد) ←

شیفت $1.F_{min}$ به سمت راست به اندازه $|E_B - E_A|$

۳) مانتیس صحیح و اعشار هر دو عدد جمع زده شود (و یا تفریق) { جمع شده اندازه علامت }

۴) نتیجه هنجار شود .

ولادت حضرت علی اکبر (ع) (۲۳۱ هـ.ق) و روز جوان

• ضرب دو عدد ممیز شناور :

$$(-1)^{S_A \oplus S_B} \underbrace{(1.F_A \times 1.F_B)}_{1.F_C} \times 2^{E_A+E_B-b}$$

الگوریتم ضرب ←

۱) چک کردن صفر

۲) نماها با هم جمع شود { جمع بسته بی علامت }

۳) $1.F_A \times 1.F_B$ محاسبه شود { ضرب بسته بی علامت }

۴) نتیجه هنجار شود . ... ، — — ← هنجار

(در صورت لزوم بایاس شیفت به سمت راست و ۱+ کردن نما هنجار می شود) .

مثال :

$$\begin{array}{rl} & +1.001 \times 2^{+2} \\ + & -1.000 \times 2^{+2} \\ \hline & 0.001 \times 2^{+2} \end{array}$$

Under flow سرریز
0.0 ...

• تقسیم دو عدد ممیز شناور :

$$(-1)^{S_A \oplus S_B} \frac{1.F_A}{1.F_B} \times 2^{E_A-E_B+b}$$

الگوریتم تقسیم ←

۱) چک کردن صفر

۲) $E_A - E_B + b$ ← در پایان

یاکسان کتاشت ساجنا سامون مروارید یاکسان ایران به پخش فراصرفه

۳) $S_A \oplus S_B$

۴) $1F_A \div 1F_B$

$1.F_A 00\ldots0 \times 2^{x} = 0.1F_A 0\ldots0 \times 2^{x+1}$

اگر $F_A < F_B$ ← صحیح

$F_A \geq F_B$ ← F_A را به سمت راست شیفت می‌دهیم

$E_A{+}{+}$

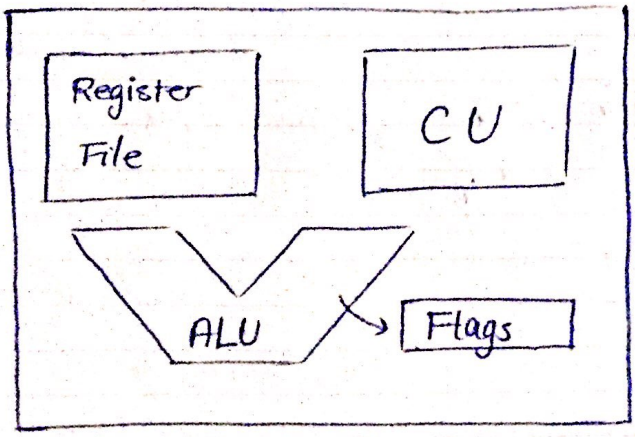
{ تقسیم شده بی علامت }

۵) $E_A \quad E_B + b$

۶) هنجارسازی نتیجه

$\overbrace{01F_A}^{m}\overbrace{00\ldots00}^{m}\ldots0 \;\Big|\; \overbrace{1.F_B}^{m}$

CPU :



به‌پخش فرامرفه آرش پاکسیان ایران مروارید سامون سیاحت کلاشت پاکسیان

• طراحی واحد کنترل :

۱- سیم بندی شده — Hardwired C.U.

۲- ریز برنامه پذیر — Micro Programmed C.U.

۲ ← انعطاف پذیری ، Debug راحت تر

۱ ← سرعت بیش تر

* طراحی واحد کنترل ← در گام دوم قرار می گیرد :

۱) طراحی مسیر داده — Data Path Design

مجموعه هر عنصری از مدار که وظیفه انتقال ، ذخیره ، بازیابی ، پردازش و ... داده ها را بر عهده دارد . مثل گذرگاه ، ALU ، Register ، ...

۲) طراحی واحد کنترل — Control Unit Design

مجموعه هر عنصری از مدار که وظیفه کنترل بدون فرآیند اجرای دستورالعمل را بر روی مسیر داده بر عهده دارد . مانند مدارات کنترلی خطوط Select MUX ها .

طراحی Software $\xrightarrow{\text{سپس}}$ Hardware

Instruction Set Architecture (ISA) $= \{I_1, I_2, \dots, I_n\}$

(مجموعه معماری دستورالعمل ها)

ISA + Size of Memory ⟶ Design

به پخش مزامیرفه آی تی پاکسان ایروان مروارید ساجنا کتاباش پاکدیان

MOV AX, BX ≡ AX ← BX

ADD AX, BX ≡ AX ← AX + BX

ADD AX, BX, CX ≡ AX ← BX + CX

POP AX ≡ AX ← Top Stack

(Stack Pointer) SP در داخل CPU ، Top را مشخص می‌کند.

Push 5

STC ≡ CF ← 1

CLC ≡ CF ← 0

[10] : خانه شماره ۱۰ حافظه

* ISA باید کامل باشد تا در طراحی CPU به مشکل نخوریم.

سؤالات برای طراحی واحد کنترل ←

۱ـ تعداد دستورالعمل‌ها

۲ـ چند نوع دستورالعمل داریم؟

۳ـ تعداد ثبات‌ها ← A : عام منظوره — General Purpose

B : خاص منظوره — Special Purpose

طراح در نظر می‌گیرد.

۴ـ CPU چند بیتی است؟ (CPU n-بیتی ≡ داده مورد نظر n بیتی است)

۵ـ فضای آدرس‌دهی حافظه؟

ولادت مشرف قائم آل محمد عجل الله تعالی فرجه (۲۵۵ هـ ق) (تعطیل) و روز جهانی مستضعفان

به‌پخش فراصفحه یاکدیان ایران مروارید سایبان کتاباش یاکدیان

۶- چه Flagهایی لازم است؟

۷- منطق نمایش داده‌ها ← صحیح و اعشاری و از چه نوعی؟

۸- دستور العمل‌ها چند عملوند دارند؟ {0, 1, 2, 3, ...}

دستورات پشته‌ای ← صفر عملوند

ADD ← دو عدد Pop می‌شوند و عمل جمع را انجام می‌دهد.

SUB

MUL

بیشتر هم روی Stack نیز

1
2
3
4

→ ((1+2) − 3) * 4

۹- نوع عملوندها ← آدرس، خود عملوند، ثبات؟

Mem. Addr.

10 , [10] , AX , [AX] , [AX+5] , [SP+SI−5]

Pointer را Support می‌کند.

immediate

Direct Reg. ← AX

inDirect Reg. ← [AX]

۱۰- شیوه‌های آدرس دهی داده؟

به نشر فراسرهه پاکسان آروان مروارید سامون سایبا کتاب پاکسان

*) شیوه‌های آدرس‌دهی داده — Addressing Modes

۱ـ بلافصل — immediate

نمود مستقیم

```
ADD AX,10
MOV BX,6
```

۲ـ ثباتی (مستقیم) — Reg. Direct

حافظه‌ای

```
ADD AX,BX
```

قبل از برنامه BX را وارد می‌کنیم.

۳ـ آدرس‌دهی حافظه‌ای مستقیم — Direct Mem. Addressing

سرعت

```
MOV [10],[5]
```

۴ـ حافظه‌ای غیرمستقیم

```
ADD [10],[[5]]
```

Pointer

۵ـ ثباتی غیرمستقیم

```
ADD AX,[BX]
```

۶ـ ضمنی

```
STC (Set Carry) , CLC
STD , CLD
```

۷ـ نسبی (با ثبات پایه)

```
ADD AX,[BX+5]
```

به‌نشر فراصرفه پاکسان ایران مروارید ساجنا کتاباش پاکسان

۸ـ شاخص دار (ثبات های خاص)

```
ADD  AX, [SI]
```

۹ـ شاخص دار با ثبات پایه

```
ADD  AX, [SI+BX]
```

DI, SI ← ثبات شاخص

*) ثبات پایه ← BP, BX

۱۰ـ خود افزایشی / خودکاهشی

```
MOVS   AX, BX
```
(move string)

```
MOVS   SI, DI
MOV    CX, 5
```
← ۵ بار افزایش می‌دهد.

به‌نشر فرامرفه آی‌کیو پاکسان ایران مروارید سامون ساجنا کتاباش پاکسان

① RISC → Reduced Instruction Set Computer

② CISC → Complex Instruction Set Computer

*) تفاوت ها

تعداد دستور العمل : ۱ـ کم ۲ـ زیاد

تعداد ثبات های داخل پردازنده : ۱ـ کم ۲ـ زیاد

تعداد طول لازم برای اجرای دستور العمل : ۱ـ معمولاً ثابت ۲ـ متغیر ـ زیاد

طول قالب دستور العمل : ۱ـ معمولاً ثابت ۲ـ متغیر

تعداد خطوط برنامه : ۱ـ زیاد ۲ـ کم

طول ثبات های داخل پردازنده : ۱ـ ثابت ۲ـ متغیر

شیوه های آدرس دهی : ۱ـ کم ۲ـ متنوع و زیاد

هزینه (مساحت) : ۱ـ ارزان ۲ـ گران

طول طلا : ۱ـ زیاد ۲ـ کم

توان مصرفی : ۱ـ کم ۲ـ زیاد

در مورد سرعت اجرا نمی‌توان حرفی زد.

Program : Compiler / Assembler → Linker → Loader

```
main (){
    char x,y,z;
    x=5;
    y=2;
    z=(x+y)*x-y+z;
    print(z);
}
```

Compiler

برنامه پذیر نمی باشد!

```
SEGMENT  para public 'data'
    X  DB ?
    Y  DB ?
    Z  DB ?
END SEGMENT

SEGMENT  para public 'code'
    MOV  X,5
    MOV  Y,2
    ADD  AX,X,Y
    MOV  BX,X
    MUL  BX
    SUB  Y
    ADD  Z
    MOV  DX,AX
    MOV  AH,5
END SEGMENT
```

AX	AH	AL

پاکسان کشاورز ساجنا موبایل پاکسان ارزان بهینش

قابل تبدیل به کد اسمبلی

Machine Language

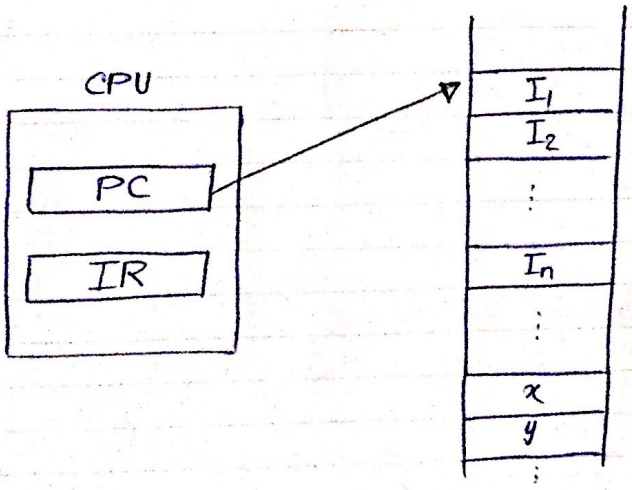
*) هر خط تبدیل به یک خط بایتی می‌شود.

قالب دستور العمل :

OpCode	Arg 1	Arg 2

کد منحصر به دستور — آدرس حافظه

EXE



PC : Program Counter
IP : Instruction Pointer
IR : Instruction Register

*) الگوریتم فون نیومن :

۱ـ واکشی دستور العمل (Fetch)

روز منابع دستی
به‌نخش عرصه‌ها یاکسان اروان مروارید سلمون ساحا کتاب یاکسان

Instruction Cycle

۲- رمزگشایی دستورالعمل — Instruction Decode

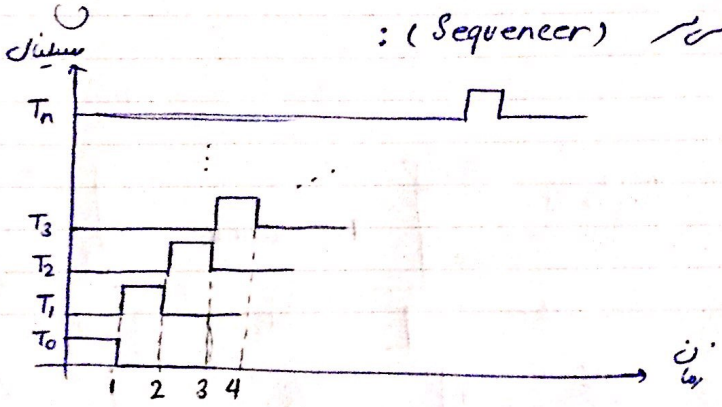
۳- خواندن عملوندها — Operands Fetch

۴- اجرای دستورالعمل (ALU) — Instruction Execute

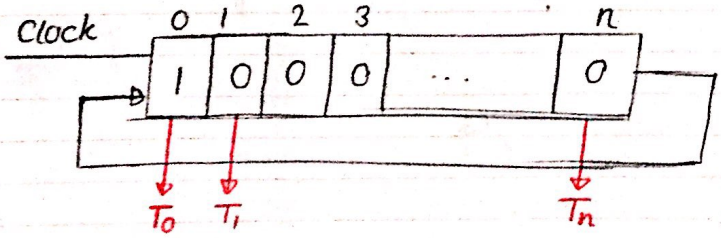
۵- ذخیره‌سازی نتایج (ob در CPU، ob در RAM) — Result Write Back

۶- PC++ و برویم!

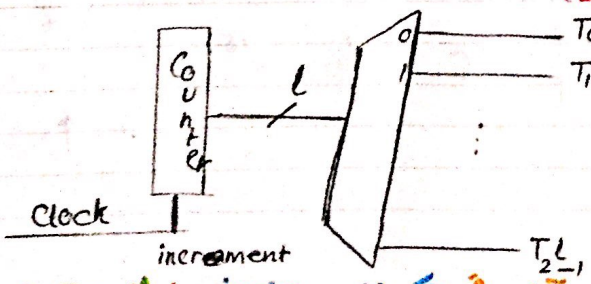
• طراحی توالی‌ساز (Sequencer):



پیاده‌سازی ۱: (بیت لغزان):



۲: (Decoder)



به‌نشر فرامرز پاکسان ایران مروارید سلمون سایتا کتاب‌شب پاکسان

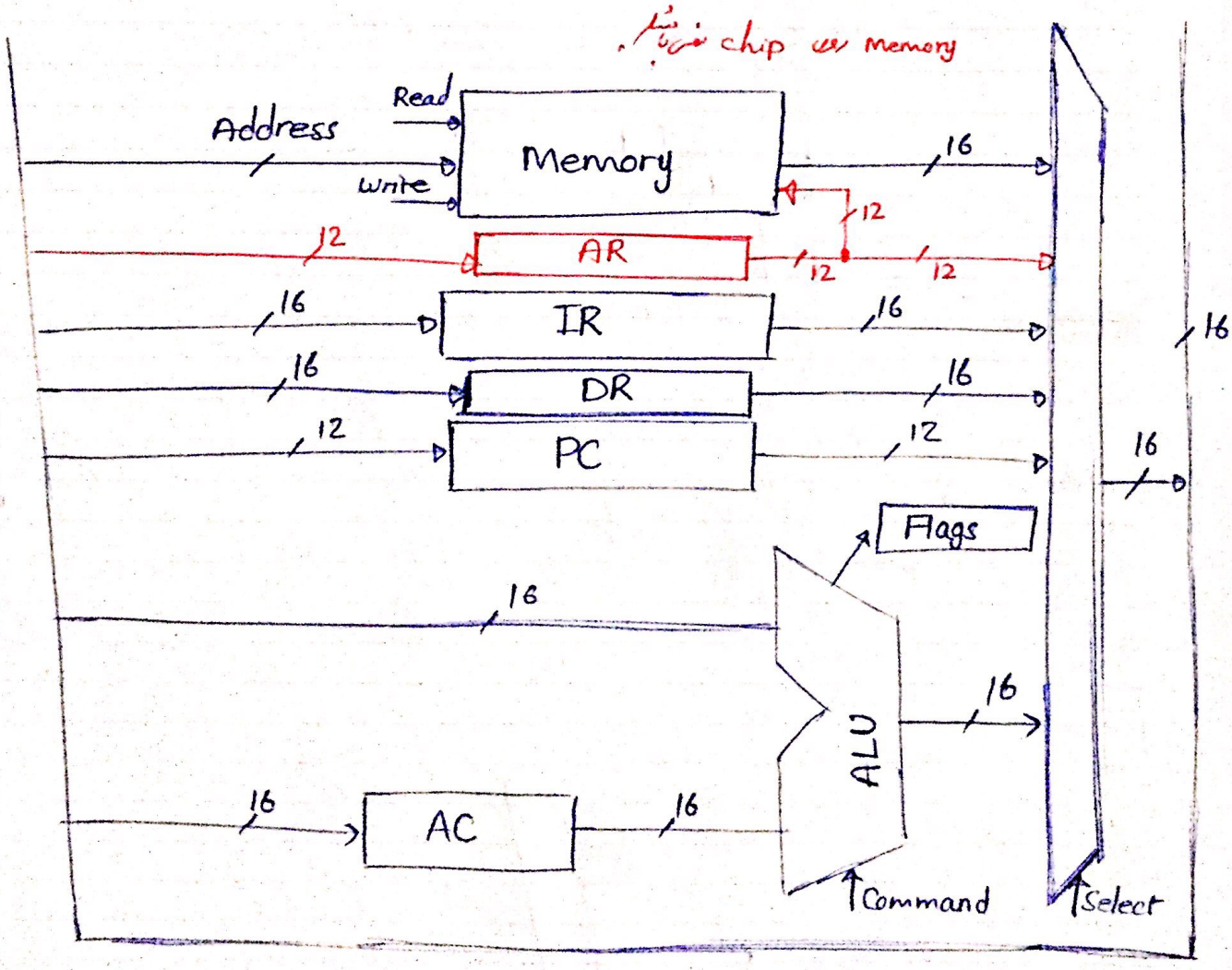
*) مهم‌ترین ایراد وارد به الگوریتم فون نیومن این است که سخت افزار را نمی‌تواند اعمال موازی با هم انجام دهد، مجبور به اعمال ترتیبی می‌کنیم.

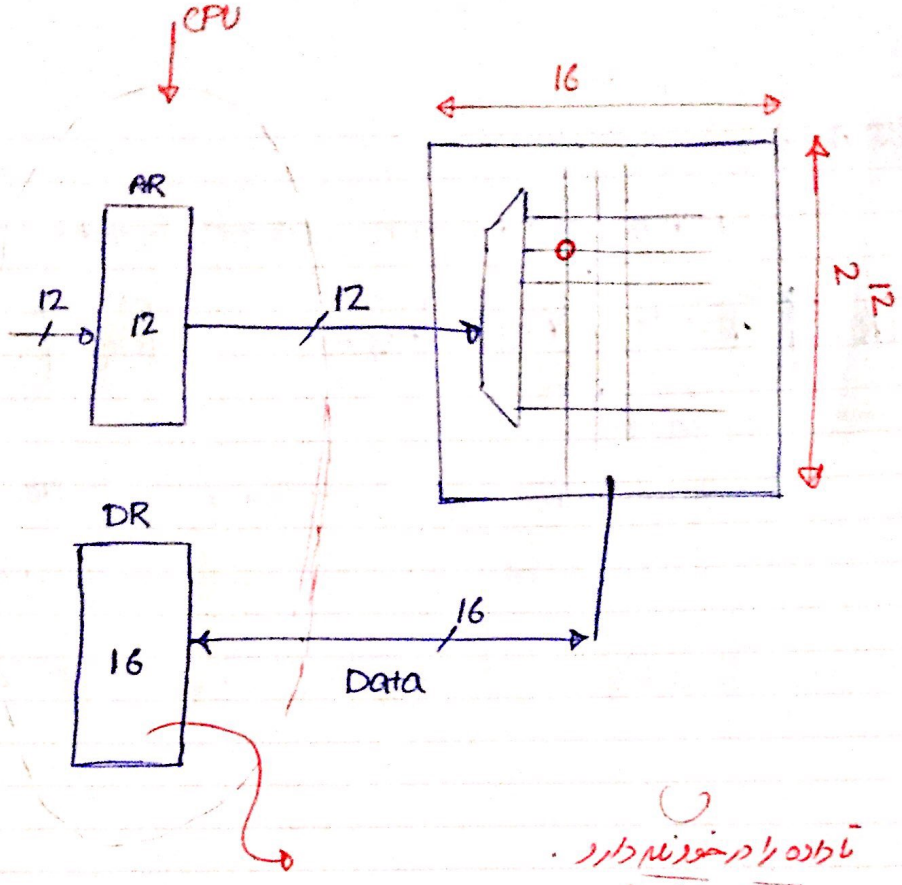
*) BUS (اتوبوس)

- تعداد سیم‌ها را روی Board کم می‌کند.

- به جای طراحی point-to-point مورد استفاده قرار می‌گیرد (خروجی به ورودی)

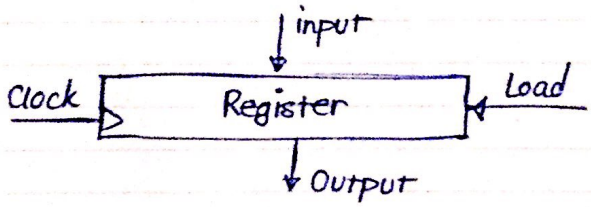
- اگر یک عنصر روی آن مقدار بنویسد، عنصر دیگری برای نوشتن روی آن باید صبر کند.



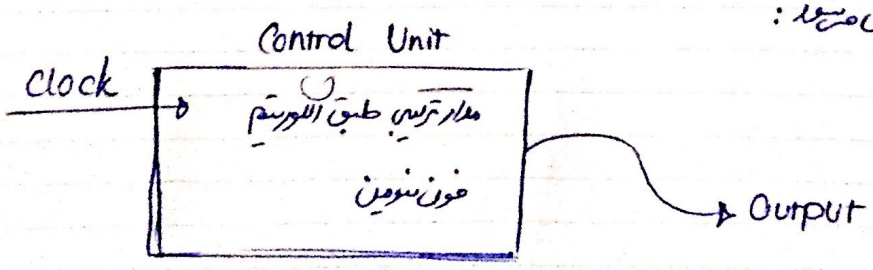


تا داده را در خود نگه دارد.
سیم این کار را نمی‌کند.

*) تمامی Register های صفحه قبل به شکل زیر هستند:



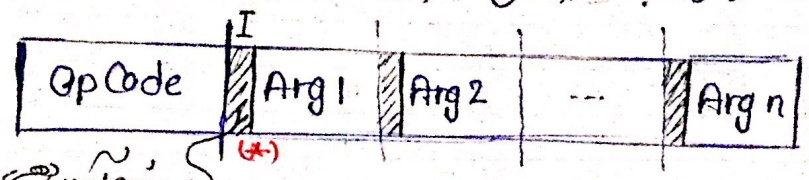
تمامی ورودی های Register های صفحه قبل و select MUX و ... به مدار زیر وصل می‌شوند:



*) BUS Arbiter :

• طراحی قالب دستور العمل (Instruction Format) :

IR :



شیوه آدرس دهی (مستقیم و غیرمستقیم)

هرچه IR ↓ ← فضای لازم برای Instruction در M.M ↓

(*) می‌تواند از جنس علا، ثبات، آدرس حافظه باشد.

(*) IR بهتر است به مضربی از اندازه word حافظه اصلی باشد.

• طراحی واحد کنترل :

۱ـ سیستم مبنی شده Hardwired C.U.

۲ـ ریز برنامه پذیر Micro Programmed C.U.

۲ ← انعطاف پذیری ، Debug راحت تر

۱ ← سرعت بیشتر

*) طراحی واحد کنترل ← در گام دوم قرار می گیرد :

۱) طراحی مسیر داده Data Path Design

مجموعه هر عنصری از مدار که وظیفه انتقال ، ذخیره ، بازیابی ، پردازش و ... داده ها را بر عهده دارد . مثل لترها ، ALU ، Register ، ...

۲) طراحی واحد کنترل Control Unit Design

مجموعه هر عنصری از مدار که وظیفه کنترل سیگنال فرامین اجرای دستورالعمل را بر روی مسیر داده بر عهده دارد . مثلاً مدارات کنترلی خطوط Select MUX ها .

Software $\xrightarrow{\text{سیس}}$ Hardware طراحی

Instruction Set Architecture (ISA) $= \{I_1, I_2, \dots, I_n\}$

(مجموعه معماری دستورالعمل ها)

ISA + Size of Memory ⟶ Design

Scanned by CamScanner

08 MOV AX, BX ≡ AX ← BX

09 ADD AX, BX ≡ AX ← AX + BX

10 ADD AX, BX, CX ≡ AX ← BX + CX

11 POP AX ≡ AX ← Top Stack

(Stack Pointer) SP در داخل CPU ، Top را مشخص می‌کند.

12 Push 5

13 STC ≡ CF ← 1

14 CLC ≡ CF ← 0

15 [10] : خانه شماره ۱۰ حافظه

16 * ISA باید کامل باشد تا در طراحی CPU به مشکل نخوریم.

17

سؤالات برای طراحی واحد کنترل ←

18

۱ـ تعداد دستورالعمل‌ها

19

۲ـ چند نوع دستورالعمل داریم؟

20

۳ـ تعداد ثبات‌ها ← A : عام منظوره B : خاص منظوره

General Purpose Special Purpose

طراح در نظر می‌گیرد.

۴ـ CPU چند بیتی است؟ (CPU n-بیتی ≡ داده مورد نظر n بیتی است)

۵ـ فضای آدرس‌دهی حافظه؟

ولادت حضرت قائم عجل الله تعالی فرجه (۲۵۵ هـ ق) (تعطیل) و روز جهانی مستضعفان

Scanned by CamScanner

۱۳۹۴
پنجشنبه
۱۴
خرداد
THURSDAY
4 Jun 2015
۱۶ شعبان ۱۴۳۶

۶ـ چه Flagهایی لازم است؟

۷ـ منطق نمایش داده‌ها ← صحیح و اعشاری و از چه نوعی؟

۸ـ دستور العمل‌ها چند عملوند دارند؟ {0,1,2,3,...}

دستورات بیشتر ای ← صفر عملوند

ADD ← دو عدد PoP می‌کند و عمل جمع را انجام می‌دهد.

SUB

MUL

بیشتر سر هم روی Stack زیر

1
2
3
4

→ ((1+2) − 3) * 4

رحلت حضرت امام خمینی (ره) رهبر کبیر انقلاب و بنیانگذار جمهوری اسلامی ایران (۱۳۶۸ هـ ش) - انتخاب حضرت آیت الله امام خامنه‌ای به رهبری (۱۳۶۸ هـ ش) (تعطیل)

۱۳۹۴
جمعه
۱۵
خرداد
FRIDAY
5 Jun 2015
۱۷ شعبان ۱۴۳۶

۹ـ نوع عملوندها ← آدرس، خود عملوند، ثبات؟

Mem. Addr.

10 , [10] , AX , [AX] , [AX+5] , [SP+SI−5]

Pointer را Support می‌کند.

immediate

Direct Reg. ← AX

inDirect Reg. ← [AX]

۱۰ـ شیوه‌های آدرس دهی داده؟

قیام خونین ۱۵ خرداد (۱۳۴۲ هـ ش) (تعطیل) - روز جهانی محیط زیست

Scanned by CamScanner

08 Addressing Modes *) شیوه‌های آدرس‌دهی داده

09 immediate ۱- بلافصل

10 فبول مستقیم
ADD AX,10
MOV BX,6

11

Reg. Direct ۲- ثباتی (مستقیم)

12 حافظه‌ای
ADD AX,BX

قبل از برنامه BX را وارد می‌کنیم.

13

14 Direct Mem. Addressing ۳- آدرس‌دهی حافظه‌ای مستقیم

15 مرجع
MOV [10],[5]

16

۴- حافظه‌ای غیر مستقیم

17 ADD [10],[[5]]

Pointer

18

۵- ثباتی غیر مستقیم

19

20 ADD AX,[BX]

۶- جنبی

STC (Set Carry), CLC
STD , CLD

۷- نسبی (با ثبات پایه)

ADD AX,[BX+5]

به‌پخش فرامرفه آیتی پاکسان ایران مروارید هامون ساجنا گلشاد پاکسان
Scanned by CamScanner

۸ـ شاخص دار (ثبات های خاص)

ADD AX, [SI]

۹ـ شاخص دار با ثبات پایه

ADD AX, [SI+BX]

* ثبات پایه ← BP, BX

ثبات شاخص ← DI, SI

۱۰ـ خود افزایشی / خودکاهشی

MOVS AX, BX
(move string)

MOVS SI, DI

MOV CX, 5 ← ۵ بار افزایش می‌دهد.

Scanned by CamScanner

① RISC → Reduced Instruction Set Computer.

② CISC → Complex Instruction Set Computer

*) تفاوت ها

تعداد دستور العمل : ۱ـ کم ۲ـ زیاد

تعداد ثبات های داخل پردازنده : ۱ـ کم ۲ـ زیاد

تعداد کلاک لازم برای اجرای دستور العمل : ۱ـ معمولاً ثابت ۲ـ متغیر ـ زیاد

طول قالب دستور العمل : ۱ـ معمولاً ثابت ۲ـ متغیر

تعداد خطوط برنامه : ۱ـ زیاد ۲ـ کم

طول ثبات های داخل پردازنده : ۱ـ ثابت ۲ـ متغیر

شیوه های آدرس دهی : ۱ـ کم ۲ـ متنوع و زیاد

هزینه (مساحت) : ۱ـ ارزان ۲ـ گران

طول کلاک : ۱ـ زیاد ۲ـ کم

توان مصرفی : ۱ـ کم ۲ـ زیاد

در مورد سرعت اجرا نمی توان حرفی زد.

Program : Compiler / Assembler → Linker → Loader

Scanned by CamScanner

```
main (){
    char x,y,z;
    x=5;
    y=2;
    z=(x+y)*x-y+z;

    print(z);
}
```

↓ Compiler

برنامه پذیر نمی باشد!

```
SEGMENT  para public 'data'

    X  DB ?
    Y  DB ?
    Z  DB ?

END SEGMENT

SEGMENT  para public 'code'

    MOV  X,5
    MOV  Y,2
    ADD  AX,X,Y
    MOV  BX,X
    MUL  BX
    SUB  Y
    ADD  Z
    MOV  DX,AX
    MOV  AH,5
      :
END SEGMENT
```

AX	AH	AL

Scanned by CamScanner

قابل تبدیل به کد اسمبلی → Machine Language

*) هر خط تبدیل به یک خط باینری می‌شود.

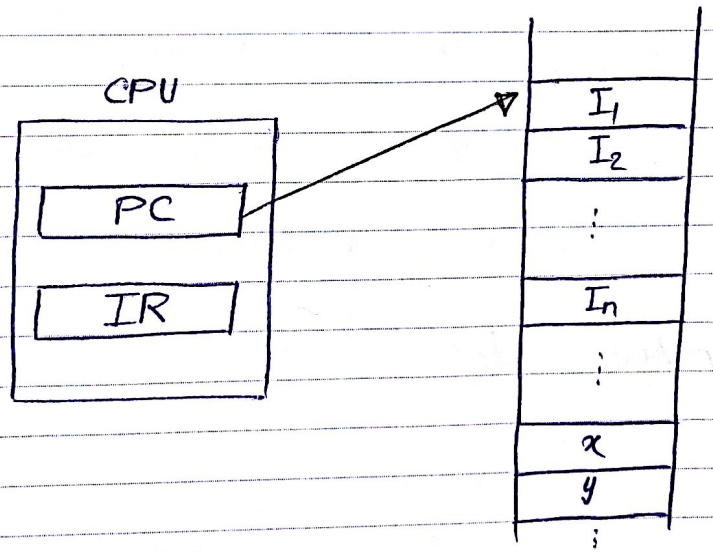
قالب دستور العمل :

OpCode	Arg 1	Arg 2

OpCode → کد مربوط به دستور

Arg 1 → آدرس حافظه

↓ EXE



PC : Program Counter
IP : Instruction Pointer
IR : Instruction Register

*) الگوریتم فون نیومن :

۱- واکشی دستور العمل (Fetch)

Scanned by CamScanner

Instruction Cycle

۲- رمزگشایی دستورالعمل — Instruction Decode

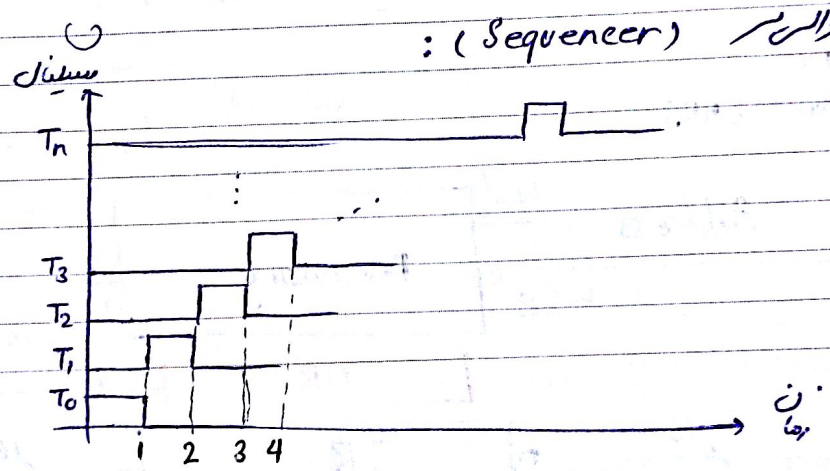
۳- خواندن عملوندها — Operands Fetch

۴- اجرای دستورالعمل (ALU) — Instruction Execute

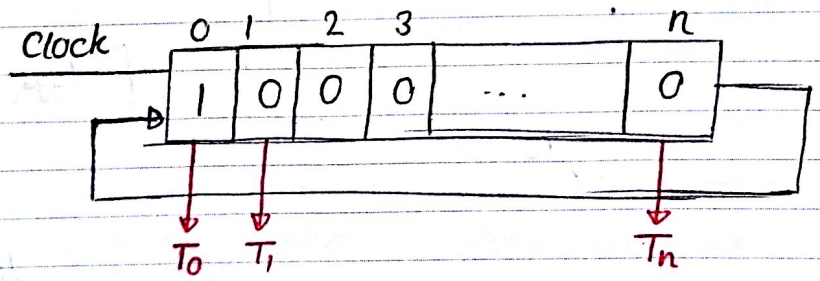
۵- ذخیره‌سازی نتایج (ob در CPU، ob در RAM) — Result Write Back

۶- PC++ و برو به ۲!

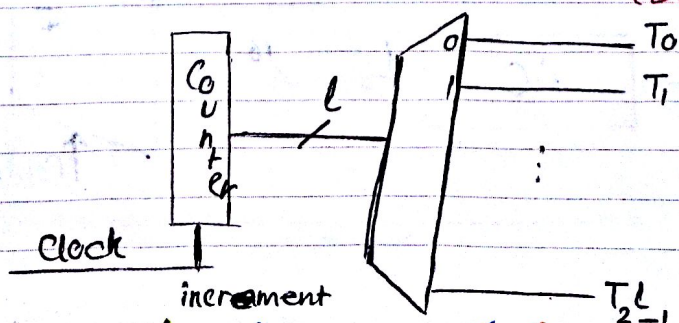
• طراحی توالی‌ساز (Sequencer):



پیاده‌سازی ۱ (بیت لغزان):



۲ (Decoder):



Scanned by CamScanner

*) مهم‌ترین ایراد وارد به الگوریتم فون نیومن این است که سخت افزار را نمی‌توانند اعمال موازی با هم انجام دهد، مجبور به اعمال ترتیبی می‌شیم.

*) BUS (اتوبوس)

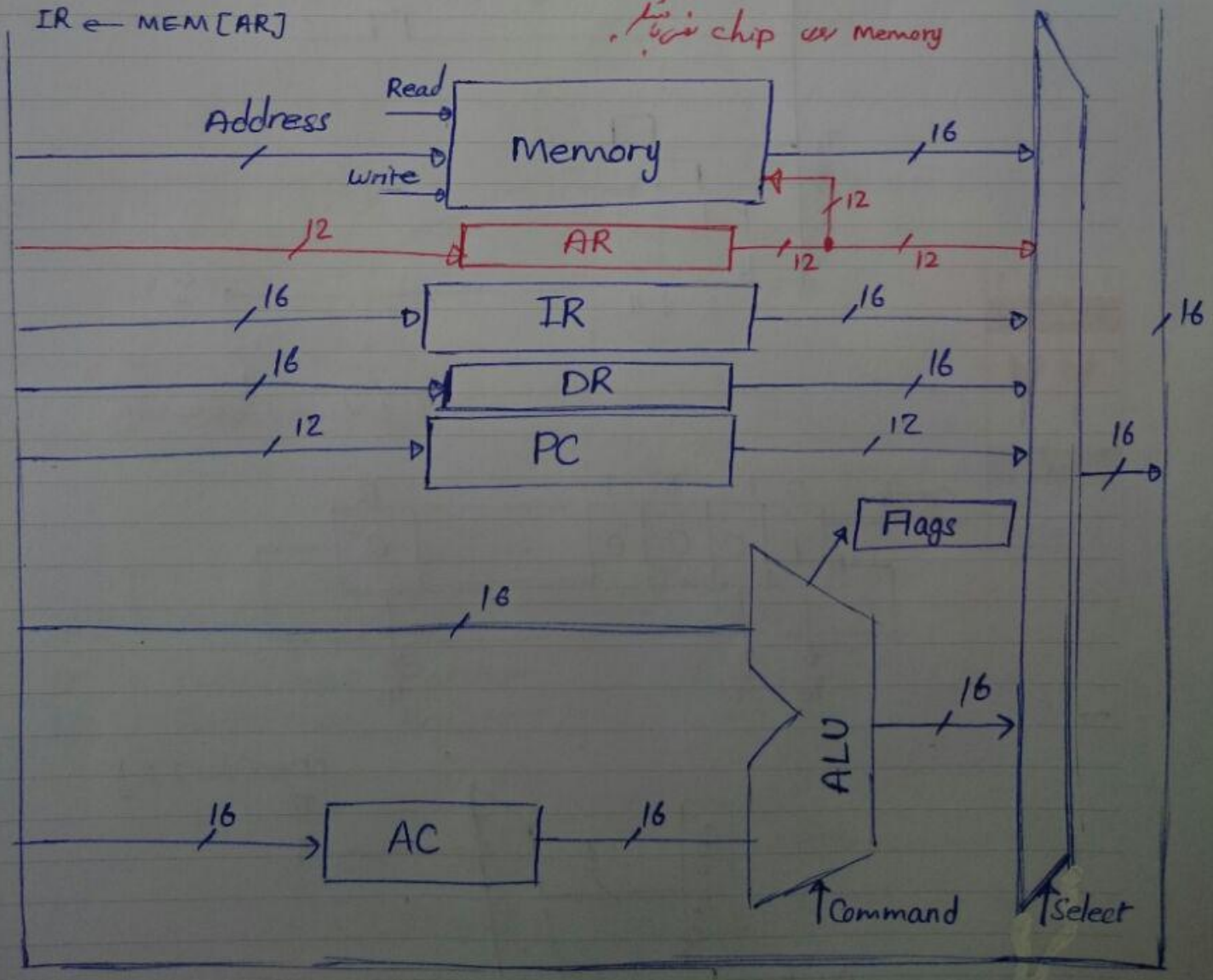
- تعداد سیم‌ها را روی Board کم می‌کند.

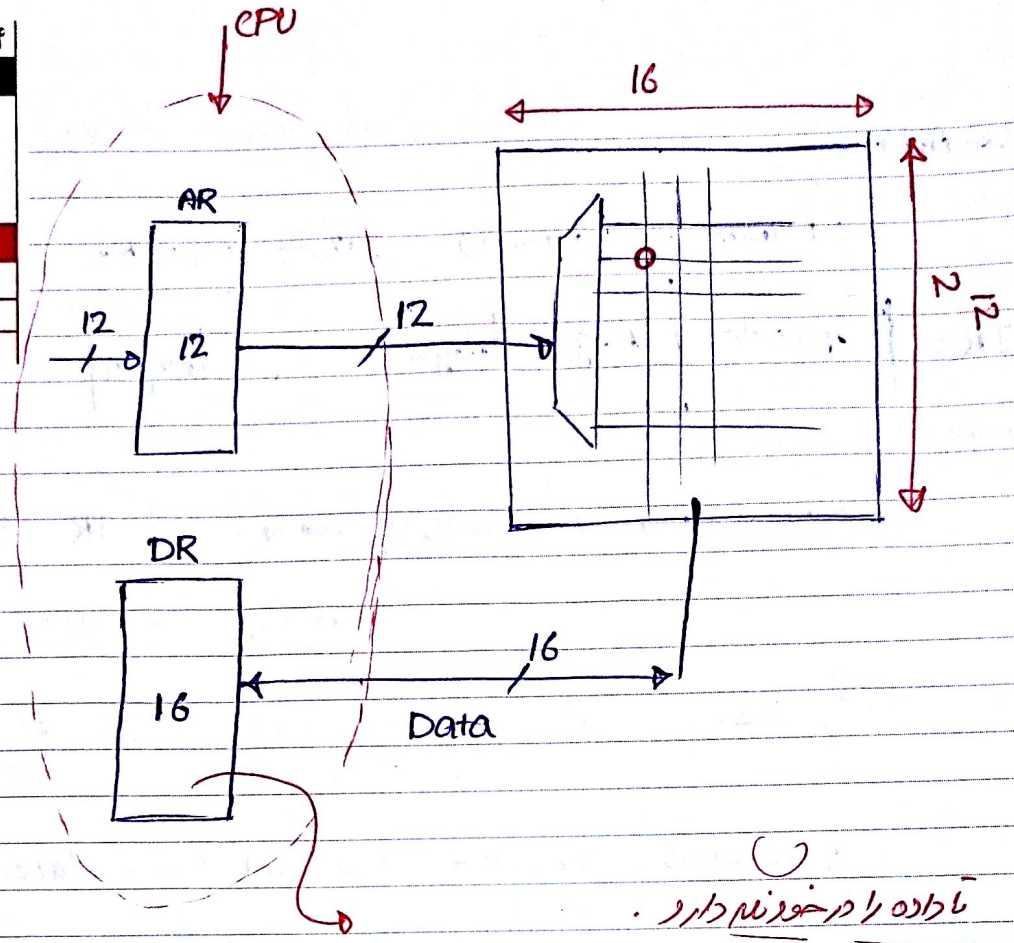
- به جای طراحی point_to_point مورد استفاده قرار می‌گیرد (خروجی به ورودی)

- اگر یک عنصر روی آن متصل بنویسد، عنصر دیگری برای نوشتن روی آن باید صبر کند.

AR ← PC
IR ← MEM[AR]

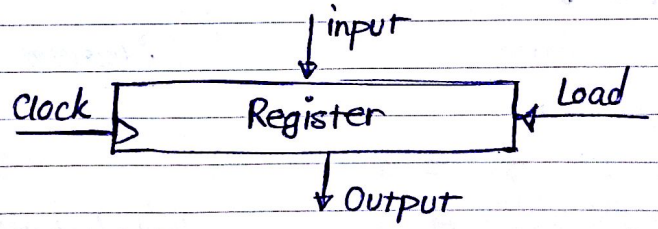
Memory روی chip نمی‌باشد.



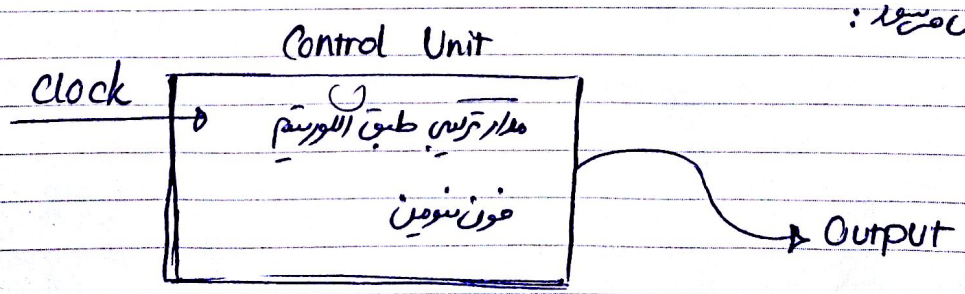


تا داده را در خود نگه دارد.
سیم این بار را نمی‌زند.

*) تمامی Register های صفحه قبل به شکل زیر هستند:



تمامی ورودی‌های Register های صفحه قبل و select MUX و ... به مدار زیر وصل می‌شود:



Scanned by CamScanner

*) BUS Arbiter :

• طراحی قالب دستور العمل (Instruction Format) :

IR :

OpCode	I	Arg 1	Arg 2	...	Arg n

(*) شیوه آدرس دهی (مستقیم و غیرمستقیم)

هرچه IR ↓ ← فضای لازم برای Instruction در M.M ↓

(*) می‌تواند از جنس علامت، ثبات، آدرس حافظه باشد.

(*) IR بهتر است مضربی از اندازه word حافظه اصلی باشد.

(*) Accumulator (انباشتگر) ← نتیجه عملیات را در آن ذخیره می‌کنیم.

*) Micro Operation (ریزعملیات) → شرط : R ← عبارتی از ثبات‌ها

• دستورهای ISA ← ۱: حافظه‌ای ۲: ثباتی ۳: I/O

• Micro Operation → ——— ; ——— ; ——— } به هم وصل هستند.

M.O با هم اجرا می‌شوند.

Scanned by CamScanner

. Performance Evaluation : (www.SPEC.org)

کارایی برای هر کس، معنای متفاوتی دارد و نمی‌توان تعریف دقیقی ارائه کرد.

Performance : ALU → Delay
Cache → hit ratio
CPU → ?

$$\text{Computer's Performance} \propto \frac{1}{\text{Execution Time}}$$

$$\Rightarrow \frac{\text{Performance A}}{\text{Performance B}} = \frac{\text{Execution Time B}}{\text{Execution Time A}}$$

$$*)\ \text{Execution Time} = \sum_{i=1}^{n} T_i = \sum_{i=1}^{n} CPI_i * \frac{1}{f} = \sum_{i=1}^{n} \frac{10^{-6}}{MIPS_i}$$

n : تعداد دستورالعمل‌های اسمبلی داخل حافظه (قابل خواندن برای CPU)

T_i : مدت اجرای دستور i ام

CPI → Clock Per Instruction
IPC → Instruction Per Clock
$\Rightarrow CPI = \frac{1}{IPC}$

MIPS → Million Instruction Per Second (1/s)

$$IPC = \frac{\#\ Inst.}{\#\ clock} * \frac{1/f}{1/f} = \frac{\#\ Inst.}{\#\ time\ (s) * 10^6} * \frac{1}{f} * 10^6$$

$$\Rightarrow IPC = MIPC * \frac{1}{f} * 10^6$$

$$MIPC = IPC * f * 10^{-6} = \frac{1}{CPI} * f * 10^{-6}$$

شهادت سربازان دلیر اسلام : بخارایی ، امانی ، صفار هرندی و نیک نژاد (۱۳۴۴ هـ.ش)

به‌نشر فراصرفه پاکسیان ایروان آرتیم مروارید هامون ساجنا گلناش پاکسیان

Scanned by CamScanner

$I_i \rightarrow CPI_i \,,\; IPC_i \,,\; MIPS_i$

$\overline{CPI} \rightarrow ? \qquad = Avg(CPI_i)$

$\overline{IPC} = Avg(IPC_i)$

$\overline{MIPS} = Avg(MIPS_i)$

$\Rightarrow ① \simeq \sum \overline{CPI} * \frac{1}{f} \simeq n.\overline{CPI}.\frac{1}{f}$

$= n.\frac{1}{\overline{IPC}}.\frac{1}{f} = n.\frac{1}{\overline{MIPS}}.10^{-6}$

$\overline{MIPS} \rightarrow MIPS \qquad \overline{IPC} \rightarrow IPC \qquad \overline{CPI} \rightarrow CPI$

*) قانون آمدال: چنانچه f درصد یک برنامه را n برابر تسریع کنیم، میزان تسریع کل برنامه از رابطه زیر به دست می‌آید:

$$\text{Speed up} = \frac{1}{(1-f) + \frac{f}{n}}$$

مثال: در یک کامپیوتر که در آن ۲۰ درصد برنامه‌ها را دستورات ممیز شناور تشکیل می‌دهند، برای افزایش سرعت یکی از راه‌حل‌های زیر انجام شده است. کدام سریع‌تر است؟

۱) دستورات ممیز شناور ۲ برابر تسریع شده است.

۲) همه دستورات ۵ برابر تسریع شده ولی طول کلاک ۱۰٪ افزایش یافته است.

Scanned by CamScanner

مثال: در یک کامپیوتر برای افزایش سرعت، عملیاتی زیر انجام شده است. میزان تسریع چقدر است؟

تعداد دستورات ۱۰٪ افزایش یافته است ولی فرکانس ۰٫۳ برابر کاهش یافته و مقدار CPI نیز ۴۰٪ کاهش یافته است.

$$\underset{1.1}{n} \cdot \underset{0.6}{CPI} \cdot \frac{1}{\underset{0.7}{f}}$$

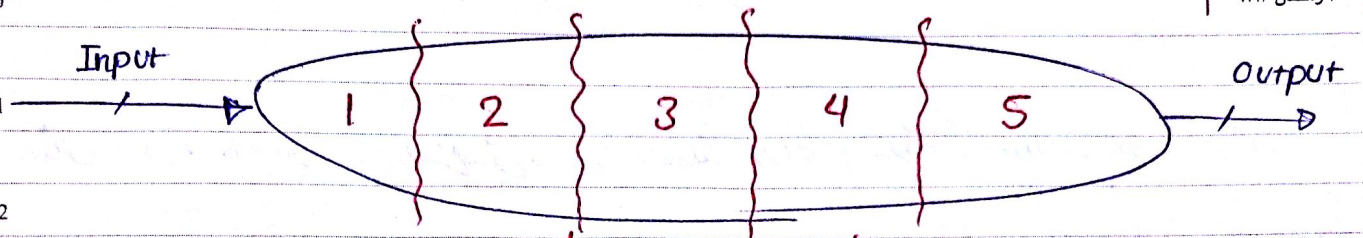
$$\text{Speed Up} = \frac{7}{6.6} > 1$$

درگذشت دکتر علی شریعتی (۱۳۵۶ ه ش)

Scanned by CamScanner

*) خط لوله (Pipe line) :

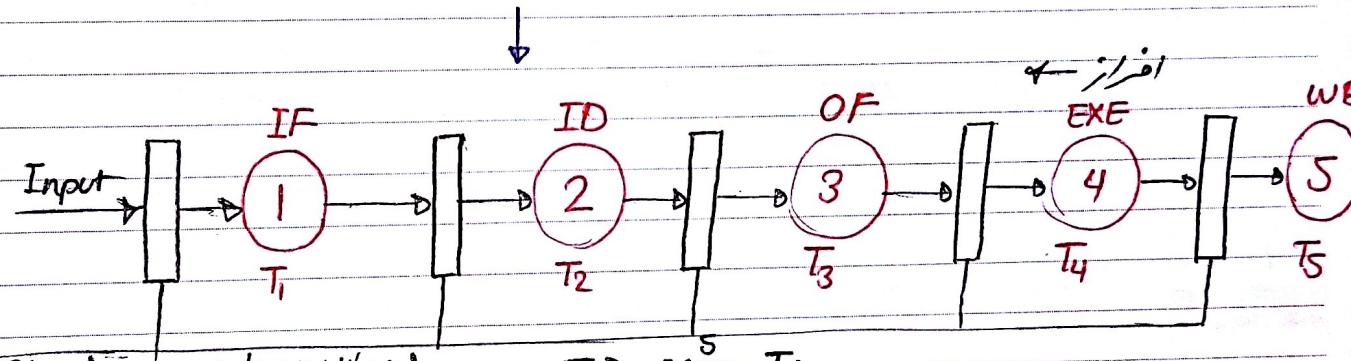
product اولی مقداری طول می‌کشد تا تولید شود ولی product های بعدی سریع‌تر تولید می‌شوند.



(Critical Path) تأخیر مسیر بحرانی : T

برای پاسخ دهی به n ورودی مختلف: nT

5_Stage (Step) PiPe line



Clock (حساس به سطح) ولی فرقی نمی‌کند.

$$\tau \geq \max_{i=1}^{5} T_i$$

$$\sum_{i=1}^{5} T_i = T$$

$$\text{تأخیر کل برای پاسخ } n \text{ ورودی} = \underbrace{5.\tau}_{\text{اولی}} + (n-1)\tau$$

$$\text{میزان سریع} = \frac{\text{واحد بلند}}{\text{واحد بلند}} = \frac{n.T}{5\tau + (n-1)\tau} = \frac{n}{5+(n-1)} \left(\frac{T}{\tau}\right)$$

$$\frac{\sum_{i=1}^{k} T_i}{\max(T_i)}$$

$5 \to k$

Scanned by CamScanner

$$k\tau + (n-1)\tau$$

$$n \to \infty \Rightarrow k$$

علاوه بر افراز مکانی، افراز زمانی متقارن نیز انجام دهیم.

(Space_Time Diagram

2	3	4	5	6	7	8
I_2	I_3	I_4				
I_1	I_2	I_3	I_4			
	I_1	I_2	I_3	I_4		
		I_1	I_2	I_3	I_4	

Scanned by CamScanner

یکی از مراحل متوقف شود ← Bubble

تا زمانی که برای ورودی جدید منتظر می‌مانیم ← Stall (توقف)

* در بعضی دستورها، دستور بعدی لزوماً خط بعدی نیست (مثلاً if) ← مشکل پیش می‌آید. (Bubble)

و این مشکل را هنگام decode می‌فهمیم، نه در ابتدای دستور.

دستور jump، دو حباب در مقدار تولید می‌کند. در if، jmp شرطی پیش می‌آید یک حباب بیشتر تولید می‌کند.

• مشکلات خط لوله در پردازنده:

- Branch Difficulties (مشکلات انشعاب)

I_1 SUB R_A, R_B
I_2 JNZ ELSE
I_3

JMP END

500 → ELSE:

END:

I_1	I_2			I_{500}	
	I_1	I_2			I_{500}
		I_1	I_2		
			I_1		
				I_1	

راه حل ←
- ایجاد Bubble
- با یکی از مسیرها داخل خط لوله پر شود
اگر غلط بود ← undo

- BTB (Branch Target Buffer): وابستگی بین Branchهای قبلی (ده دستور اخیر JMP را نگه می‌دارد)
• در if اولی then اجرا شود، در if های بعدی هم به احتمال زیاد then اجرا می‌شود

Branch Predictor
~99% hit

روز تبلیغ و اطلاع رسانی دینی (سالروز صدور فرمان حضرت امام خمینی (ره) مبنی بر تاسیس سازمان تبلیغات اسلامی - ۱۳۶۰ ه ش) - روز اصناف

Scanned by CamScanner

- Resource Sharing (Conflict)

مثلاً دو دستور به ALU نیاز داشته باشند ← Bubble

مثلاً IF ، OF هر دو به BUS نیاز دارند.

راه حل ← Resource بیشتر پول بیشتر

- Data Dependency

I_1 : $R_2 \leftarrow 3$

I_2 : $R_1 \leftarrow R_2 + 5$

راه حل ← Instruction Reordering (جابجایی دستورها)

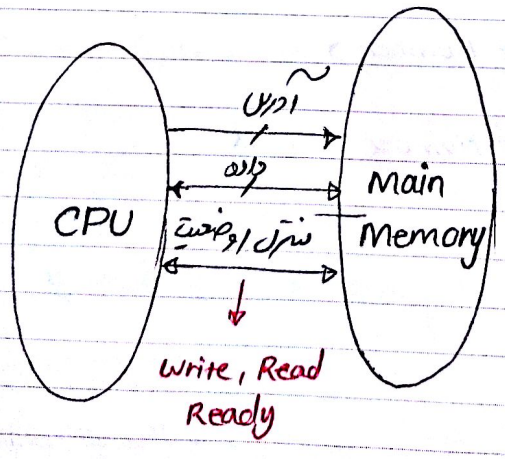
ایجاد Bubble

*) سازمان ورودی / خروجی :

۱- وجود واحدی به نام رابط interface در دستگاه های IO

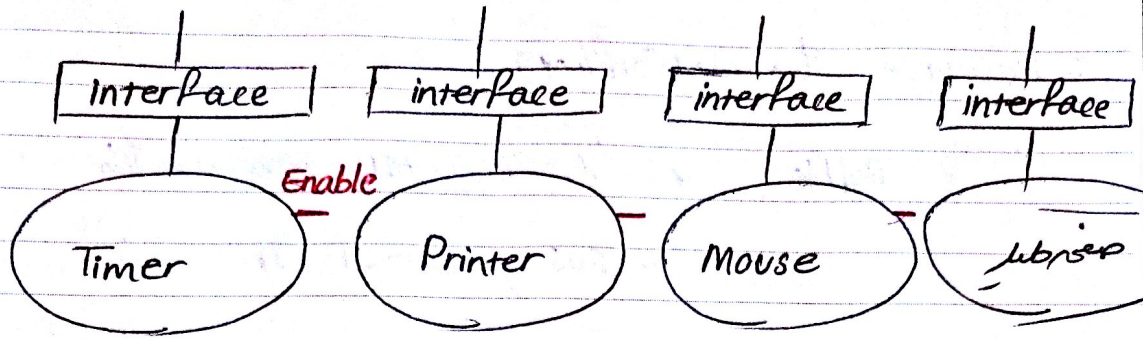
۲- شیوه های اتصال IO به CPU

۳- DMA



ارتباط با دنیای بیرون؟

Scanned by CamScanner



وظایف interface :

۱- تنظیم کردن سطوح ولتاژ برای دستگاه‌های مختلف

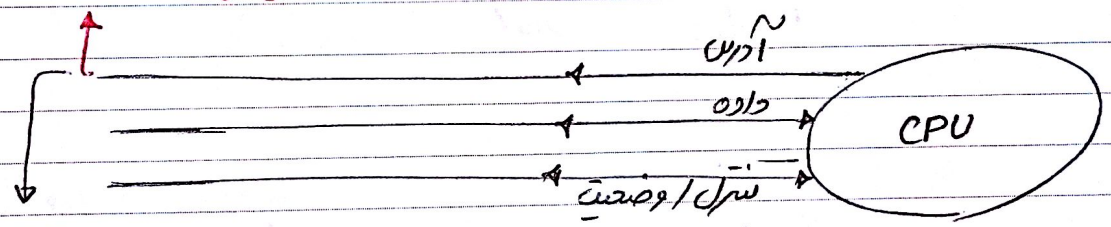
۲- یکسان سازی قالب داده‌ها

۳- سرعت انتقال IO (می‌تواند نقش Buffer را هم داشته باشد).

۴- Buffer کردن داده‌ها

برای این PN به پایه Enable دستگاه‌ها متصل می‌شود.

شیوه‌های اتصال IO به CPU :



برای مشخص کردن این پایه

interface نام

interface

(Port Number ← برای هر دستگاه unique)

(IN) input AX, < Port Number >

(OUT) output < Port Number > , AX

۱- Memory با لترهای متفاوت و IO با لترهای متمایز از Memory به CPU وصل شود.

(isolated IO)

مشکل: ساخت پایه‌های زیاد برای CPU سخت است.

Scanned by CamScanner

۲- Memory و IO ها در بلندرها آدرس، داده، کنترل/وضعیت یکسان باشند. (Memory Mapped IO)

مشکل: Conflict داده‌های Memory و یکی از IO ها ← می‌سوزد.

می‌توان با تغییر ولتاژ به یکی از این دو اولویت داد.

• معادل بودن Read/Write مربوط به M.M با input/output مربوط به IO ها ← یکی از ترفندها ...

- هدر رفتن n خانه از M.M (n: تعداد IO ها)

۳- Memory و IO در بلندرها آدرس و داده مشترک ولی در بلندرها کنترل/وضعیت متفاوت هستند.

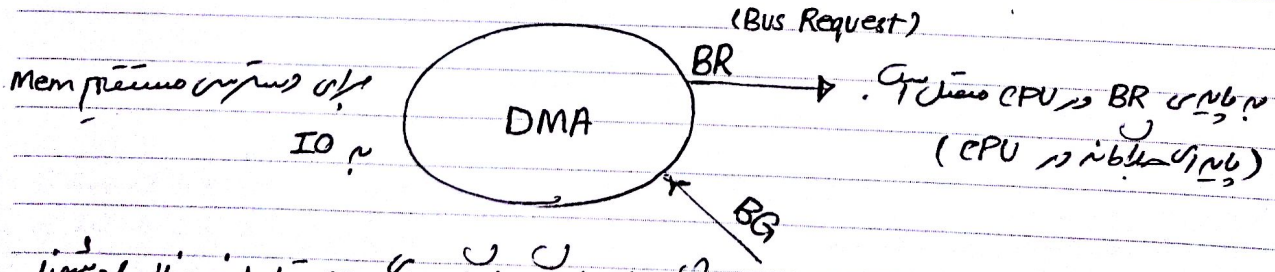
به جای دو خط کنترل/وضعیت فقط یک خط IO/M به CPU اضافه می‌شود تا به سازنده می‌دهد دستور مربوطه، باید به M.M برود یا IO.

• تأخیر مربوط به دستگاه‌های IO بسیار زیاد است.

• DMA (Direct Memory Access):

تنها عنصری است که به جز CPU در برخی موارد فرماندهی را بر عهده می‌گیرد.

DMA به هر سه بلندرها آدرس، داده و کنترل/وضعیت متصل است.



از CPU می‌گیرد ← هرگاه BG شود، بلندرهای CPU قطع (غیرفعال) می‌شوند.

← همزمان دستور R/W به Mem، IO می‌رود.

Scanned by CamScanner

. در هنگام درخواست :

- DMA ، BR را یک می‌کند .

- DMA ، آدرس شروع را به CPU اطلاع می‌دهد .

- DMA تعداد بایت مورد نظر را گزارش می‌کند .

DMA خودش یک IO است .
از طریق برنامه می‌فهمد که باید BR را یک کند

(برای تضمین بازگشت فرمان‌ها)

. شیوه‌های انتقال اطلاعات در DMA :

۱- Burst Transfer (انتقال یک باره)

CPU برای مدتی کار نمی‌کند .

۲- Stealing Cycle (سیکل دزدی)

یک Transfer ← یک instruction در CPU ← transfer ← ...

Scanned by CamScanner

*) انواع انتقال اطلاعات :

۱- سری

۲- موازی

سرعت موازی بیشتر است ولی احتمال خراب شدن بیت ها بیشتر است. امروزه روش سری بیشتر مورد استفاده قرار می گیرد.

۱- همعام (سنکرون) در لحظه های زمانی ، فرستنده و گیرنده با هم تطابق دارند. (توافق)

۲- ناهمعام (آسنکرون)

امروزه اکثر ارتباطات ناهمعام از نوع سری است (فرستنده یا گیرنده یا هر دو کلاک ندارند).

به جز CPU بقیه ناهمعام هستند.

CPU با Memory ← ناهمعام موازی

CPU با IO ← ناهمعام

اگر پهنای بیت سری را افزایش دهیم ← موازی

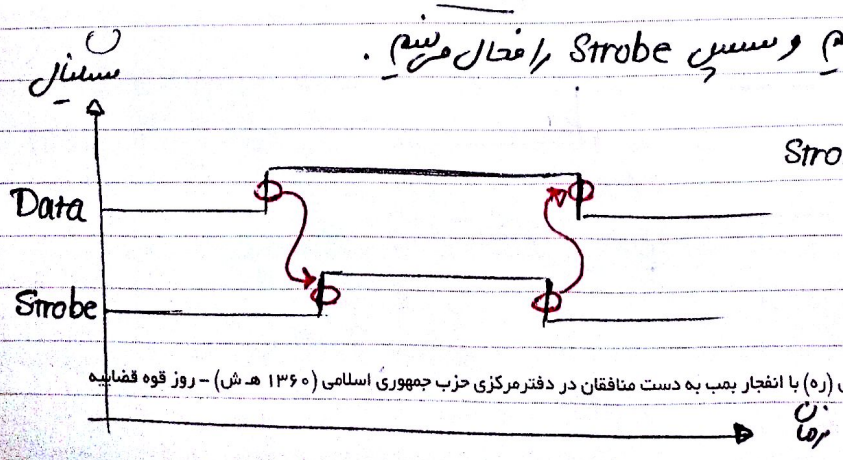
حالت های داده ← صفر ، یک ، علم آمادگی برای انتقال داده

۱- Strobe دار ← A : با شروع فرستنده B : با شروع گیرنده

۲- دست تکانی Handshaking ← A : با شروع فرستنده B : با شروع گیرنده

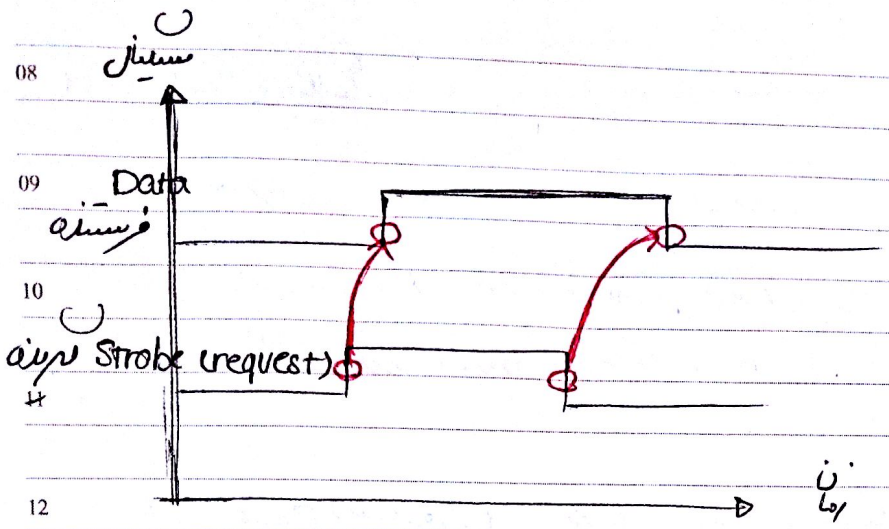
A-۱ : اول داده را آماده می کنیم و سپس Strobe را فعال می کنیم.

در هنگام پایان انتقال نیز ابتدا Strobe را خاموش می کنیم.

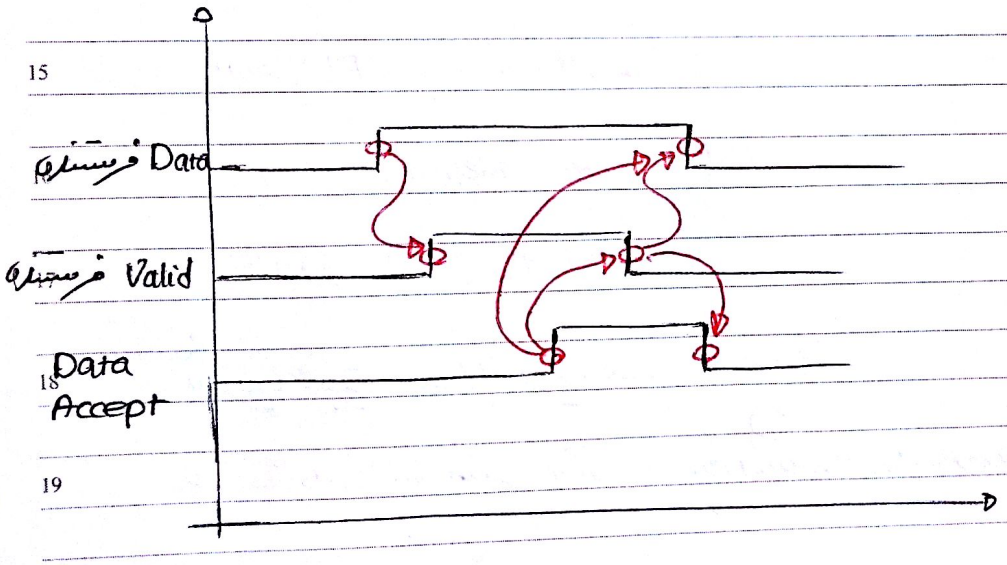


Scanned by CamScanner

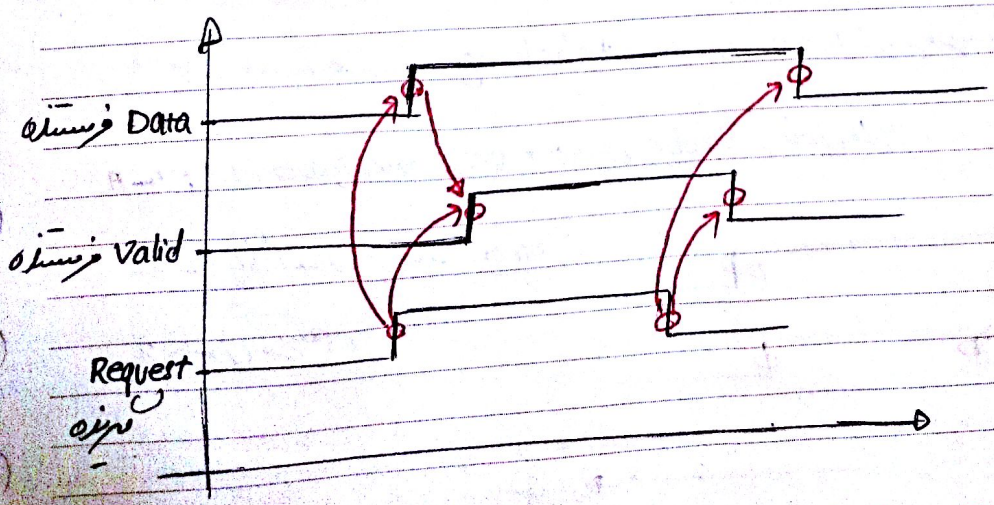
B-۱ :



A-۲ :



B-۲ :



روز مبارزه با سلاح های شیمیایی و میکروبی

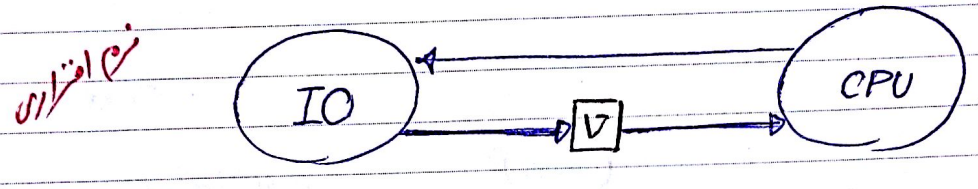
Scanned by CamScanner

- CPU با Memory ← ناهمگام ، Handshaking به اعتبار گیرنده (read)
به اعتبار فرستنده (write)

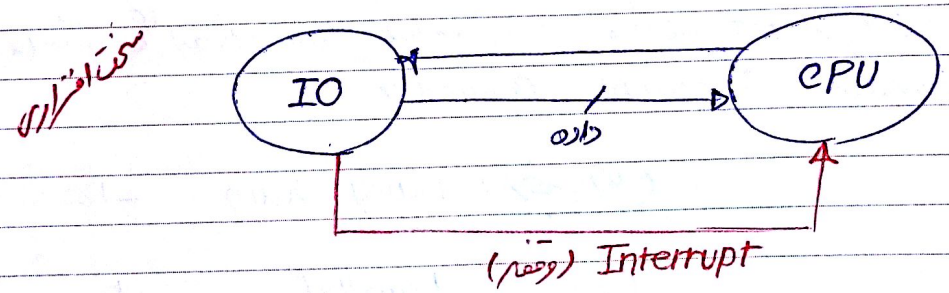
*) مشکل‌های دریافت اطلاعات بین CPU و IO :

Cache نمی‌توان نداشت (علم وجود همبستگی زمانی و مکانی) ولی Buffer می‌توان نداشت.

۱- polling (سرکشی)



۲- Interrupt (وقفه)



برنامه‌ای که CPU در هنگام وقفه انجام می‌دهد ← Interrupt Routine (برنامه وقفه)

همه وقفه‌ها حتماً برنامه وقفه دارند.

۱- Vectored (سوار)

هنگامی که دستور وقفه را می‌فرستد، آدرس روتین را نیز روی BUS داده بلافاصله به CPU اعلام می‌کند (هنگامی که CPU دستور INTA را فعال می‌کند، IO آدرس روتین را می‌فرستد).

آدرس روتین روی حافظه بورد مشخص شده که با IO، And می‌شود.

آدرس ثابت است.

به‌نشر فراصرفه پاکسبان آتی‌فم پاکسبان سامون ساجنا کتاباش پاکسبان
Scanned by CamScanner

۲- Non_Vectored (غیر سرواری)

آدرس روتین وقفه از روی یک جدول (Interrupt Vector) برداشته می‌شود.

M.M

0	Interrupt No.1
1	Interrupt No.2
	⋮

وقفه‌ها:

۱- سخت افزاری (پایه‌ها)

۲- نرم افزاری (RST)

وقفه‌ها اولویت بالایی دارند.

فقط فعال شدن با هم فرق دارد.

*) شیوه‌های اتصال چندین سیگنال وقفه IO به یک پایه INT از CPU:

PIC : Programmable Interrupt Controller

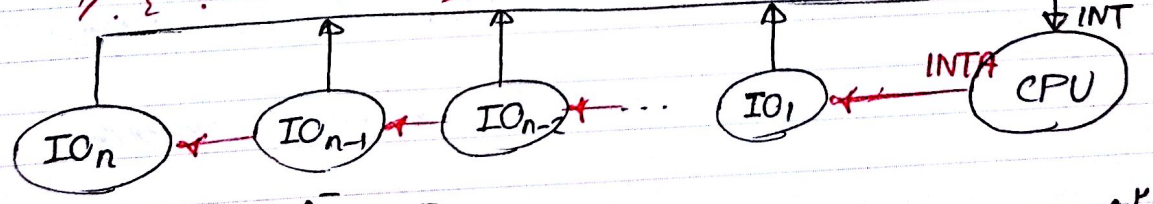
۱- Daisy chain (زنجیره‌ای)

۲- Parallel (موازی)

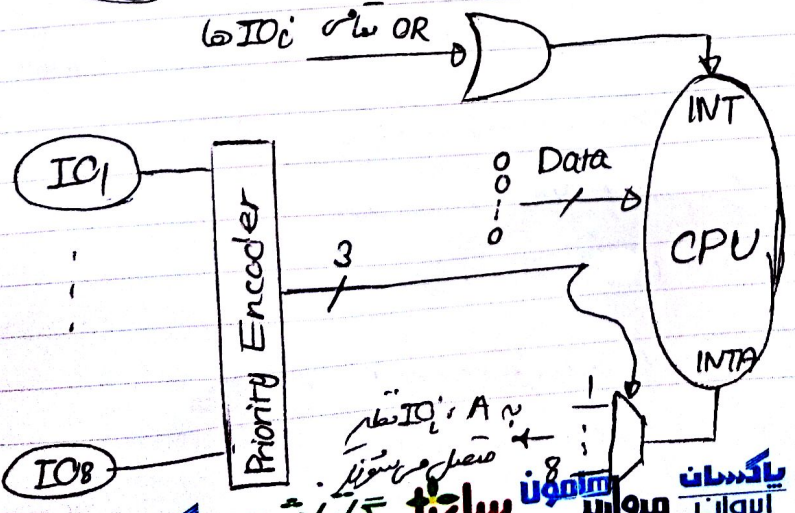
در IOها خیلی مهم
هم حساس به لبه است و هم
حساس به سطح.

خطر حساس به لبه ← Noise (عوض شدن بیت) ⇐ حساس به سطح بهتر است.

۱:



۲:



روز صنعت و معدن
به‌پخش
فرا صرفه
آتیه
پاکسان ایروان
مروارید
سامون
ساپنا
گلشاد
پاکسان
Scanned by CamScanner

۱۳۹۴
پنجشنبه
۱۱
تیر
THURSDAY
2 Jul 2015
۱۵ رمضان ۱۴۳۶

- CPU یک Flag به نام EI دارد که با دستور INT ، AND می‌شود که باعث می‌شود CPU در حلقه‌ی INT ها غرق نشود.

شهادت آیت الله صدوقی چهارمین شهید محراب به دست منافقان (۱۳۶۱ هـ ش) ولادت حضرت امام حسن مجتبی (ع) (۳ هـ ق) و روز اکرام

۱۳۹۴
جمعه
۱۲
تیر
FRIDAY
3 Jul 2015
۱۶ رمضان ۱۴۳۶

حمله ددمنشانه ناوگان آمریکای جنایتکار به هواپیمای مسافربری جمهوری اسلامی ایران (۱۳۶۷ هـ ش)

Scanned by CamScanner