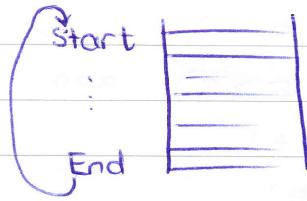


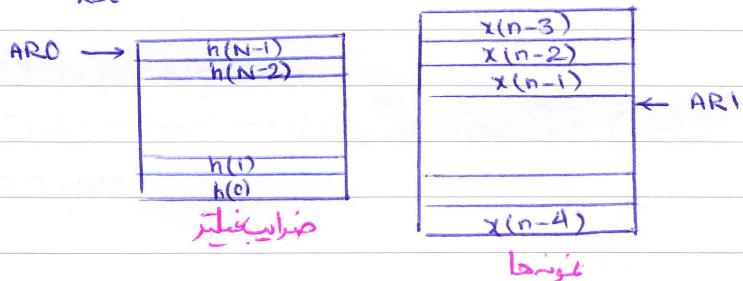
\* AR n++ (IRO)  $\xrightarrow{\text{?}} \text{Bit Reverse Addressing}$   
 $\xrightarrow{\text{?}} \text{Circular Addressing}$



## FIR : Filter Implementation

$$y(n) = \sum_{k=0}^{N-1} h(k)x(n-k)$$

الاستناد الاربعيني



Fetch

Decode

Read

Execute

الخطوات الأربع طرها لا تتوافق در حل سطح انجام دهن زمان يلي سطح خطا زياره (ns).



حالات دلیل این است که در حل سطح يلي سطح خطا زياره (ns).

حالات متعارض این است که این واحدها مواردی تقدیر مدارک داشته باشند است درهم.

برای اینکه فرطانی طریق پیدا نمایند بالا رود از pipeline استفاده می شود.

Pentium 9 levels in pipeline

:

Pentium4 20 levels in pipeline (بالآخر = performance)  
 با بالا برد و بعد از طبقات فرطان بالآخر است

نه تنون خلیم تعداد طبقات را بالا برد. بلکه علت اینکه تعداد طبقات خلیم نشسته تواند زیاد باشد این است که در  
 دستور jump با branch اتفاق بینه طرها IDE

Input → R

R ← R \* 1.5

Comp ( R, 3 )

Jump ( R &gt; 3 ) BBB

R ← R + 1

Sqrt ( R )

Output R

BBB:

Pipeline ( cache level ) احتمالی تعداد کارهای pipeline را می‌دهد.

نحوه داشتن سرطان jump برقرار است یعنی باید این jump برداشته شود ( بدین معنی وارد pipeline نمی‌شود ) طبق از سطوح امداده شود. هرچند طبقات pipeline بسیار پیچیدت باشند همچو.

## Boot Loader:

پردازنده boot loader ( بطوریکه الگوریتم تراویث از سطح USB یا دیسک را در خود دارد ) که اینجا CPU را بروز رسانی می‌کند.

## Branch Operations:

- 1) standard branch
- 2) Delayed branch
- 3) Branch prediction

- Standard branch: Empty pipeline before performing branch ( BR )

- Delayed branch : Doesn't empty pipeline ( BRD )

3) متوجه شد که branch را از اینجا که

حالت فعلی pipeline را در خود داشته باشد.

LDF \* + ARI(5), R2

BGED SKIP ↳ conditional delayed branch

اجرا	LDFN R2, R1
می‌خواهد	SUBF 3, 0, R1

; Dummy operation to complete delayed branch

MPYF

:

SKIP: LDFN

- Branch prediction :

در pentium از این روش استفاده نمی‌شود.

از متوجه شد که branch را پیش بینی نمی‌کند.

میترنلر level : DSP  
 pipeline delayed branch syntax : Pentium  
 برای آن میانک انت در جواب را بدان  
 برای آن میانک انت در جواب را بدان

$$14 \text{ MFLOPS} = 14 \text{ میلیون عمل اتم (flop)}$$

با سرعت نهادن افت نهادن performance

## Code Composer Studio

- Texas Instruments

Simulation : ساخت افزار احتساب نهادن

Emulation :

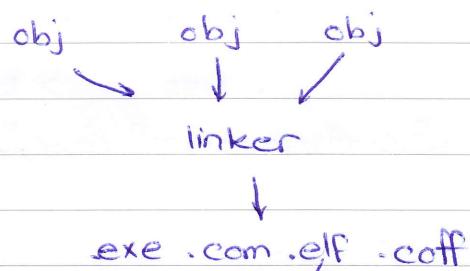
In-system-programmable

لارم نست علدرو میورا از مدار خارج دین و وارد programmer دین به برایانه وصل است و برایانه آر (Ar) میورد و برای زده تنهام میتاین میت زیرا توپک توپک میورد و را دین و رویی همچوک را برایانه بینند.

Emulation : شبیه اسی / اصراع برنامه داخل مدار (دقیق تراز سیستم)

در طالیک و فضایی بینانه ای خالی کنند معمم با راه از assembly ای تقاده نمود (نهادن C). السی روز بروز و راه ای هوش تر میگردند و برنامه های compiler ای خالی میگردند (از قل) ای خالی میگردند.

C → Compiler → Assembly → Assemble → opcode (object)



linker : بین برایانه (برایانه) برایانه میانگینه کند و رجیو فایل های را ای نام بگذار

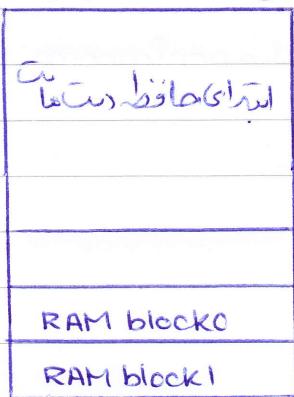
ایستقایا برایانه را به assembly بنویس : بخوبی استقایا و یخ میگیری (دارد و سرعت بسته است).  
 ای ده loop را باید کسریت ای ای ای برایانه معمم باشد (تغییر iteration ها را دیگر کنید)،  
 این طریق را بگویی.

CPU : Microprocessor Mode

= boot loader/Jhaf

## Microcomputer Mode ( Boot Loader Mode)

## Microprocessor Mode:



سیستم میزبان دو ماہ فرق دارد. Memory Map

Boot loader  
Mode =



```
int main (void)
{
    int *a = (long *) 0x809c10;
    *a = *a * 2;
}
```

## Linker Script:

Specify the Section's Allocation into Memory:

## Code

.text : > RAM0

## Initialization Tables

.cinit : > RAMO

## Constants

.const : > RAMQ

## Systems stack

. stack : > RAM

## Variables

bss : > EXT, block 0x10000

نیز قسمت از حافظه برای مواردی است که نیاز به تخصیص داده نمی‌گردد.

Jul 05 text file is

لودز:

## : Linker Script

RAM0 0x74000000 0x1000

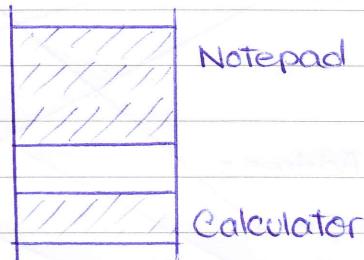
RAM1 0x86000000 0x1000

## EXTRAM OXO

.text EXTRAM  
 .stack RAM0  
 .data RAM1  
 .const RAM1

## Memory Management Unit:

برای از قدرت های حافظه ای مختلفی استفاده کرد.



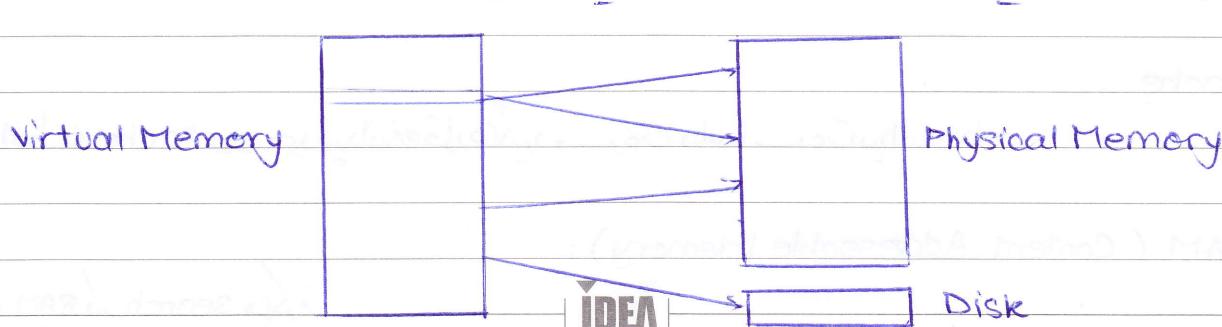
برای debug نوتن پد از آدرس مجازی (0x1001000) استفاده می شود. application آدرس های فیریم را نگه دارد و از ظرف خود میگذرد. آدرس های مجازی دارنده طرف دست است. این آدرس های مجازی را در آدرس های فیریم تبدیل کند. MMU این آدرس های مجازی را در debug نوتن (خطاباتی) خواهد نمود (آدرس هادر). آدرس های مجازی در نسخه دیدرهم به عنوان دوست زیرا application مسخوات از فضای ایجاد می شوند.

## Multitasking Kernel:

- Multiple processes at the same time
- Each process needs its own part of the memory
- Libraries :
  - ~ Static library
  - ~ Shared library

## Virtual Memory :

بنابراین محدودیت مجازی و حافظه فیریم وجود ندارد.



## Paged Virtual Addressing:

- We divide the memory and virtual memory into pages.
  - We map the pages together (not each address)

## Page Table :

Process 1:	Virtual Address	Physical Address
	0x100 ****	
	0x101 ****	
	0x102 ****	

Page Table:	Virtual Address	Physical Address
Process 2:	0x100 ****	Block 1 (Shared)
	0x101 ****	Block 2 (Shared)
	0x102 ****	

block is shared (Shared library)

استریکچر اسسیس (ویرایشی) RAM (پیغام دهن). (این خیلی بار page table، process، این را در اینجا در داده از دسترس نموده است.)

Page table can be multi-leveled.

## Translation Lockaside Buffer:

روی خود پردازند. page table می تواند مورد استفاده قرار دلیل پذیرش روی آن قرار دارند. برای مثال با این داده، (TLB) نوعی cache است

## Cache :

حافظه cache روی خود سر دارند و تراویح کنند و مرعات دیر از پای راه را در دارند.

CAM (Content Addressable Memory):

روابطی CAM ایجاد کنند (Search) آدرس فیزیکی آدرس مجازی را در دیانته و آدرس فیزیکی آدرس مجازی را در داده (آدرس فیزیکی جست).

TLB هم تواند (و تایپ است) آدرس های Instruction / Data (Harvard معملاً) را در یافته باشد.

A Typical TLB:

- Size: 4096 entries
- Hit time: 1 clock cycle (مقدار بینه زند)
- Miss penalty: 100 clock cycle (آن آدرس مجازی در TLB نباشد)
- Miss rate: % ۱ (مقدار آدرس مجازی در تLB هست)

X86 Protected Mode

(Real Mode: آدرس فیزیکی و آدرس مجازی یکسان نمی‌شوند) application ها فقط آدرس مجازی را می‌بینند.

- Memory paging
- Larger physical

غیر فردار: hard disk , flash  
فردار: RAM

سريع برآز RAM : flash , harddisk

harddisk ↔ RAM ↔ CPU

باعث افزایش سرعت

(ادهاد دستور العمل های کم به آن می‌بیند اصلاح داریم)

ذری و سرعت کم : DRAM

کوچک و دران : SRAM

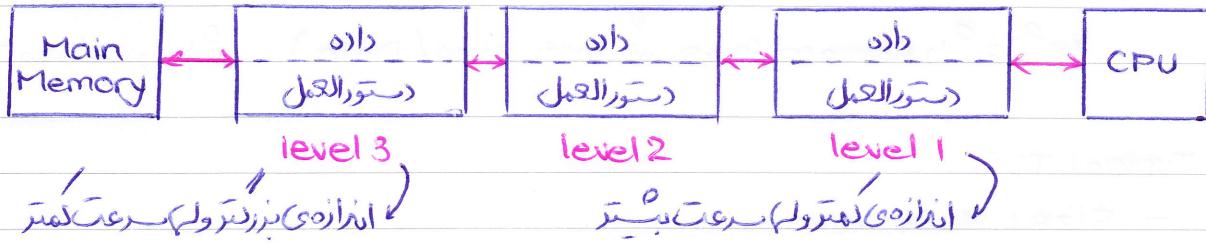
harddisk ↔ DRAM ↔ SRAM ↔ CPU

(حجم کم طبقه بعدها (نتیجه DRAM هست) می‌بینید است)

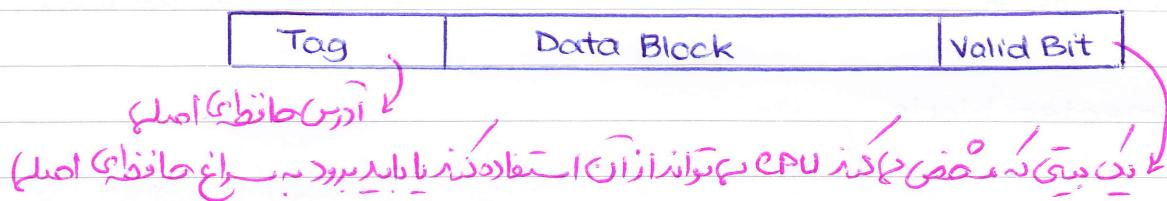
استفاده از SRAM هم برای read و write استفاده می‌شود. (معنی انتقال اطلاعات و دادهها از دو طرف)

## Cache چیز

حافظه اطلاعاتی را که در حافظه از حافظه ای level اول داریم. زمان (سترسی) به حافظه طبقی دستکاری شده است.



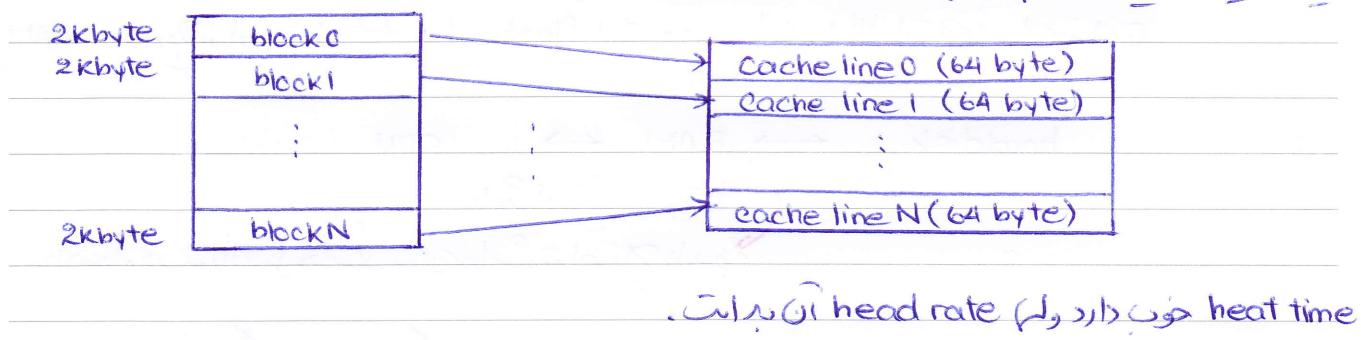
= Cache ~~überlappen~~



ابتدأ Cache بـ CPU (سترس) حيث اخذت ورقة search انها مطلوبة. الارادات موردة تظر رايانه  
داده CPU ذاته بـ وارده به صافحة اصلها عراجه بمفرد ودرجها في Cache (غيره ساکن وتخفيه  
نمود (تا دو خاتم بعد سرعان میتر داشته باشد)

## Direct Mapped Cache

آن تکمیل نهادن زمان جت و جو را (اردو لی) عملدرد آن از همه بیرون است.



## Fully Associative Cache:

معکاری طانی حافظه در هر چانه (لصومه حافظه میران) ذخیره شود و یعنی عملکرد را در دولم ساختار این (TLB) بینهایت heat time خوب است ولی heatrate Cache

## 2 Way Set Associative

بررسی عمل مراجع main, cache(), map() فقط در داده‌ها

(40 ورقة)

TLB

(Cache IM)

Data

Cache

Instruction

Access SRAM, DRAM (الوصول إلى)

SRAM:

مداد دايركت دايم باس Data روی آن باقی ماند.

DRAM: (Synchronous)

این طریق سازی Refresh داریم.

SRAM → Asynchronous

↓ Synchronous

سازی بایلی ساعت دارد.

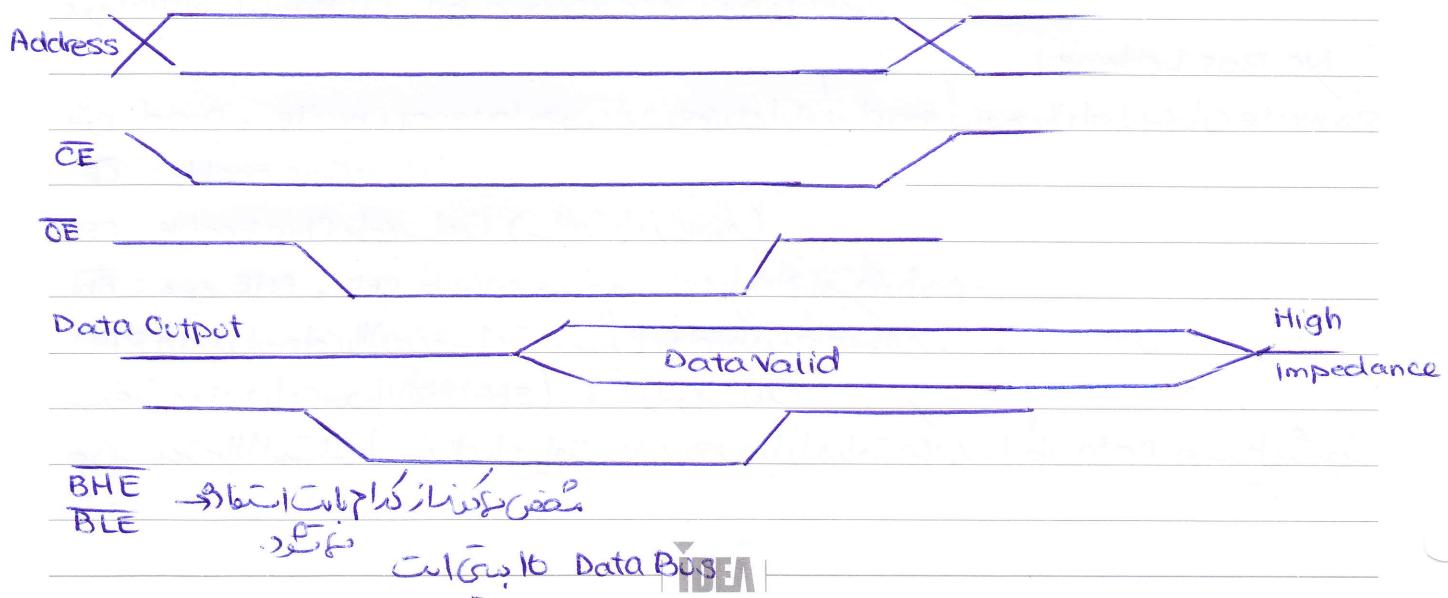
SRAM: وسترن 10ns بایت طلی نمکی در قیمت بالای \$100  
پایه های دارد آدرس رام صفحه جدا شده.

16Bit دیتا باس

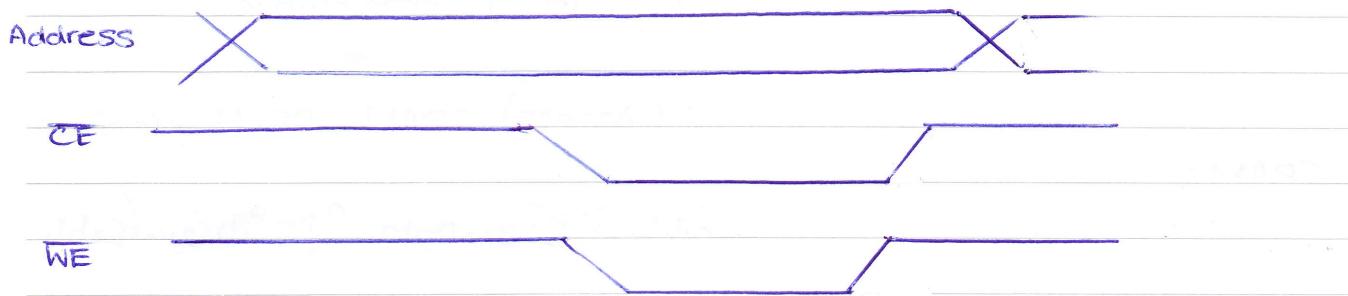
{ آدرس کی ناچیز پیغام داده و بخطاب (High-Z) خروجی دارد که این انتقال را ممکن نموده است

Chip Enable: CE  
Active Low(Output Enable): CE است زمانی وقتی است که خروجی درست نمود.  
(Write Enable): WE: باعث نمودن باشندگان خروجی بینیم

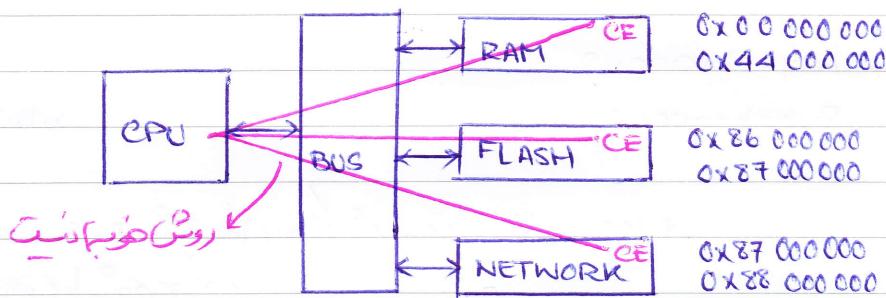
Timing Diagram Read Operation (Asynchronous)



## Async-SRAM write Cycle:



What is chip enable for?



فقط یکی از این های که تواند روی باس قرار بگیرد (در هر لحظه فقط یک نفر روی باس قرار دارد) پس CE را فعال می‌کند. (این ایجاد اختصار بسته به CPU متناسب است) همچنان CE ها را فعال کنند زیرا تنها یکی از آنها CE خود را برای CPU حاصل می‌کند (اد).

طاری بتاین است که Address باس (آدرس) قرار دهد و مثابه آدرس CE مرتبط را با خال (Decoding) کند.

## Synchronous SRAM

عنوان SRAM است فقط طرح اسکلت Clock باز می‌گیرد.

No Bus Latency:

محدودیتی ندارد بعد از آن read و write درین latency می‌گذرد.

Chip Enable: CE

Chip Enable: CEN (اعتراض بین)

عنوان BLE، BHE طاری کند و مخصوص مکانیزم RAM باشد طاری داریم.

فقط در لبهای بالارونه ساعت و سیگنال های دینامیک داشته باشد.

ساعت عملکرد را ببرد باز (266 MHz) طاری باش ساعت

جهت دریافت این سیگنال را دریافت نماید بعد دیگر از دریافت کند. انتظار Data (وسیله) می‌گیرد.

خاصیت جالب این جیب آن است که لزوماً آدرس را به آن نمایه نمود. آدرس را معمولی خواند و تولید کند.  
آدرس ۵V/LD و قابل بایان آدرس را به طور خودکار آدرس را بخواهیم بود هر دو اجازه داده از خانه های  
پیش سرهم استفاده کنیم (burstread/burst write) میزان امتحان.

## DRAM Memory:

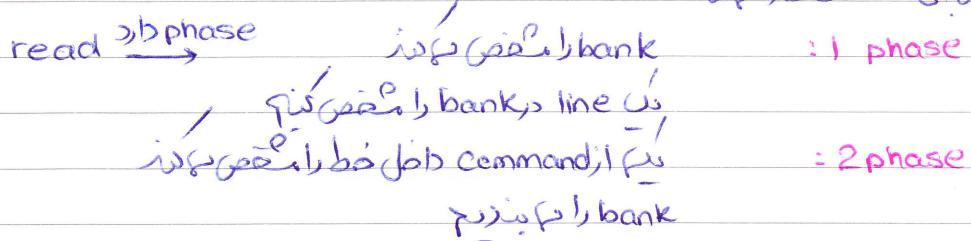
Micron MT46V64M16

Total number of bits : 16bits

4 bank

16M lines

Each line is 16 bits



## Cast Latency :

تأثیری در سرور و قرار در قرنی (دیتا رو) باس

دستابس DRAM معمولاً همان است که طول بین write و read بین ۱۰ و ۲۰ نانو ثانیه است. فرط این طرزی بالا ولی باز (ده) باس خوب کدام است (کلائم کنونی، سرعت یا مقابله کرد).

## Write Operation:

Write burst

قبل از write باید bank را فعال کنیم (آدرس مستقیم دارد: مارکی باس / مارکی خط در باس) / خارج و بخط

## Acorn:

BBC Micro by Acom

BBC advertised for BBC Micro widely

مکانیزم / CPU می تواند کامپیوٹر را باز خود ری CPU باید درست روت کریم کند و آنرا باز خود ری Acorn RISC i CPU ای Berkley می باشد

ابی CPU انقلابی بود و باعث ایجاد Acorn Risc اسپر را عو功 کرد.

## ARM2 Processor.

## Spinoff : Advanced RISC Machines

1992:

ARM6 : The result of joint efforts by ARM, Acorn and Apple