Subject: 35

* AR n++ (IRC) B → Bit Reverse Addressing
  → Circular Addressing

FIR: Filter Implementation

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

<table>
<thead>
<tr>
<th>ARD</th>
<th>ARI</th>
</tr>
</thead>
<tbody>
<tr>
<td>h(n-1)</td>
<td>x(n-3)</td>
</tr>
<tr>
<td>h(n-2)</td>
<td>x(n-2)</td>
</tr>
<tr>
<td>h(0)</td>
<td>x(n-1)</td>
</tr>
<tr>
<td>h(1)</td>
<td>x(n-4)</td>
</tr>
</tbody>
</table>

Fetch
Decode
Read
Execute

Pipeline

Pentium 9 levels in pipeline

Pentium 4 20 levels in pipeline
Subject: 3b

<table>
<thead>
<tr>
<th>Instruction</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Input → R</td>
<td></td>
</tr>
<tr>
<td>R ← R + 1</td>
<td></td>
</tr>
<tr>
<td>Comp (R, 3)</td>
<td></td>
</tr>
<tr>
<td>Jump (R &gt; 3) BBB</td>
<td>Jump pipeline</td>
</tr>
<tr>
<td>R ← R + 1</td>
<td>pipeline</td>
</tr>
<tr>
<td>Sqrt (R)</td>
<td>pipeline</td>
</tr>
<tr>
<td>Output R</td>
<td></td>
</tr>
</tbody>
</table>

Boot Loader:

### 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)

- Branch prediction:

```plaintext
LOD x + AR1(5), R2
BCED SKIP  → conditional delayed branch
```

### Example (Pentium):

```plaintext
IDFN R2, R1
SUBF 3, 0, R1
NCP   ; Dummy operation to complete delayed branch
MPYF
```

```plaintext
SKIP: IDFN
```

A branch prediction example in Pentium architecture.
Subject: CPU

<table>
<thead>
<tr>
<th>CPU: Microprocessor Mode</th>
<th>Boot Loader Mode</th>
</tr>
</thead>
<tbody>
<tr>
<td>Microcomputer Mode (Boot Loader Mode)</td>
<td></td>
</tr>
</tbody>
</table>

**Microprocessor Mode:**

<table>
<thead>
<tr>
<th>RAM block0</th>
<th>RAM block1</th>
</tr>
</thead>
</table>

**Memory Map**

- Reserved for boot loader operations
- RAH block0
- RAH block1

```c
int main (void)
{
    int *a = (long *) 0x2091c10;
    *a = *a * 2;
}
```

**LinkerScript:**

Specify the sections' allocation into Memory:

- Code: .text → RAM0
- Initialization Tables: .init → RAM0
- Constants: .const → RAM0
- Systems stack: .stack → RAM1
- Variables: .bss → EXT, block 0x1000
- System → RAM1

```
RAM0 0x74000000 0x1000
RAM1 0x76000000 0x1000
EXTRAM 0x0
```

Linker Script
Memory Management Unit:

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

Virtual Memory:
**Paged Virtual Addressing:**
- We divide the memory and virtual memory into pages.
- We map the pages together (not each address)

**Page Table:**

<table>
<thead>
<tr>
<th>Process 1</th>
<th>Virtual Address</th>
<th>Physical Address</th>
</tr>
</thead>
<tbody>
<tr>
<td>0x100 ****</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0x101 ****</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0x102 ****</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Process 2:**

<table>
<thead>
<tr>
<th>Virtual Address</th>
</tr>
</thead>
<tbody>
<tr>
<td>0x100 ****</td>
</tr>
<tr>
<td>0x101 ****</td>
</tr>
<tr>
<td>0x102 ****</td>
</tr>
</tbody>
</table>

Page Table can be multi-leveled.

Translation Lookaside Buffer:

Cache:

CAM (Content Addressable Memory):
Subject: Harvard Instruction/Data vs. TLB

A Typical TLB:
- Size: 4096 entries
- Hit time: 1 clock cycle
- Miss penalty: 100 clock cycle
- Miss rate: %.1

X86 Protected Mode
(Real Mode: Real Mode and Protected Mode application)
- Memory paging
- Larger physical

Flash, hard disk, flash, hard disk, RAM

Hard disk ↔ RAM ↔ CPU

Hard disk ↔ DRAM ↔ SRAM ↔ CPU

(Read/write access to SRAM)

استفاده از SRAM و هم چنین read و write اشکالاتی دارد.
Cache

Direct Mapped Cache

Fully Associative Cache

2 Way Set Associative
TLB

Data → Cache

( copper IM ) Instruction

( Access ) SRAM , DRAM

SRAM:

DRAM: ( Synchronous)

SRAM → Asynchronous

Synchronous

Timing Diagram Read Operation (Asynchronous)

Address

CE

OE

Data Output

Data Valid

High Impedance

BHE

BLE

Data Bus
Async SRAM Write Cycle:

Address

CE

WE

What is chip enable For?

CPU

RAM

FLASH

NETWORK

0x0 0 000 000
0x44 000 000
0x86 000 000
0x24 000 000
0x47 000 000
0x26 000 000

Synchronous SRAM

No Bus Latency:

Write, Read, Write, Read

Chip Enable: CE

Chip Enable: CEN

BLE, BHE: BW

Data

266 MHz
DRAM Memory:

Micron MT46V64M16

Total number of bits: 16 bits

4 bank

16M lines

Each line is 16 bits

Read Operation:

read \rightarrow bank

read phase: 1 phase

Each bank, line, or command: 2 phase

Write Operation:

write burst

write 1 or read \rightarrow DRAM

DRAM \& SRAM performance (طاقمنشینی از لایه

قرنطینه بالا و از دست دریافت کرده که (طاقمنشینی و یا

مکانیکی‌ی)

write bank \rightarrow (آدرس مورد نظرهاهای‌دار

bank)
Subject: Acorn

Acorn - BBC Micro

BBC Micro was advertised for BBC Micro widely.

ARM2 Processor.

Spinoff: Advanced RISC Machines

1992:

ARM6: The result of joint efforts by Acorn and Apple