

Moon-2

CS31

Pascal Van Hentenryck



Overview

Moon-2 (I)

- Multiple Cycle Implementation of Moon
- Datapath Revisited
- Breaking the instruction in cycles

Single Cycle Implementation

What is the limitation of SCI?

- The clock cycle must be as long as the longest instruction

What is the longest instruction?

- SUB
- One Rom access (reading the instruction)
- One Ram access (reading the data)
- One arithmetic operation

Assume (for simplicity)

- 50 nsec to read a ROM/RAM
- 30 nsec to subtract
- rest is negligible

How much do we lose?

Measurement Data

Instr.	Time	Freq.
LAD	100 ns.	30%
JZ	50 ns.	30%
SUB	130 ns.	15%
SAD	100 ns.	15%
LIM	50 ns.	10%

average time: 84.5

Time of an instruction on a SCI

SCI time: 130ns.

Average time with a variable clock cycle

average time: 84.5ns

Penalty

54%

→ each instruction requires 54% additional time in average

Multiple Cycle Implementation

Key Idea of SCI

- an instruction executes in 1 clock cycle

Key Idea of MCI

- Execute an instruction in several clock cycles

Advantages

- Faster

Inconveniences

- Control is more difficult
- Datapath is a little bit more complicated

Moon-2: multiple Cycle

Breaking instructions into pieces

Execution Cycle

- Fetch the instruction
- Execute the instruction

Constraints

- At most one memory operation per clock cycle (memory takes time)
- At most one arithmetic operation per clock cycle (arithmetic takes time)

Note that

- independent operations can be carried out at the same time
- example: incrementing the PC and performing a memory operation

Moon-2: Data Path

Basic Organization

- Instructions are executed in at most three clock cycles; almost all of them are executed in two clock cycles
- First cycle is always “fetching the instruction”
- Second cycle is load/store
- Third cycle is arithmetic operation if needed
- The PC is always updated in the last cycle of the instruction

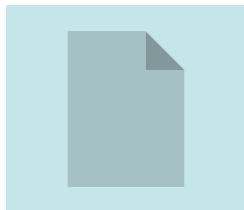
Moon-2: Datapath

Requirements

- To break an instruction into pieces we need to store intermediary results in registers

New Hardware

- An instruction register
- An arithmetic register



Moon-2: Instructions

LAD

1. Ir := ROM[Pc];
2. Acc := RAM[Ir:2];
Pc := Pc + 1;

LIM

1. Ir := Rom[Pc];
2. Acc := Ir:5;
Pc := Pc + 1;

SAD

1. Ir := Rom[Pc];
2. RAM[Ir:2] := Acc;
Pc := Pc + 1;

SOU

1. Ir := Rom[Pc];
2. Or := Acc;
Pc := Pc + 1;

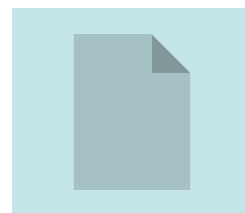
Moon-2: Instructions

SUB

```
1. Ir := Rom[Pc];  
2. Ar := RAM[Ir:2];  
3. Acc := Acc - Ar;  
  
   Pc := Pc + 1;
```

JZ

```
1. Ir := Rom[Pc];  
2. Pc := if Acc=0 then Ir:5  
   else Pc+1;
```



Performance Analysis

How to choose the clock cycle?

- Most costly operation
- reading memory: 50 ns.

Operations

LAD	100 ns.	30%
JZ	100 ns.	30%
SUB	150 ns.	15%
SAD	100 ns.	15%
LIM	100 ns.	10%

average time: 107.5 ns.

Penalty over Theoretical Best

27% additional time in average

Gain over SCI

27%