

CENG3420

Lab 3-3: LC-3b Datapath

Wei Li

Department of Computer Science and Engineering
The Chinese University of Hong Kong

`wli@cse.cuhk.edu.hk`

Spring 2020



香港中文大學

The Chinese University of Hong Kong

Overview

Introduction

Lab3-3 Assignment

Golden Results



Overview

Introduction

Lab3-3 Assignment

Golden Results



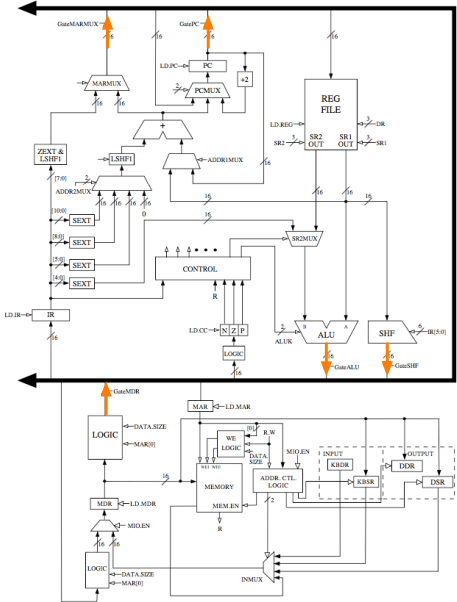
The Slides are self-contained? NO!

Do please refer to following document:

- ▶ [LC-3b-datapath.pdf](#)
- ▶ [LC-3b-ISA.pdf](#)



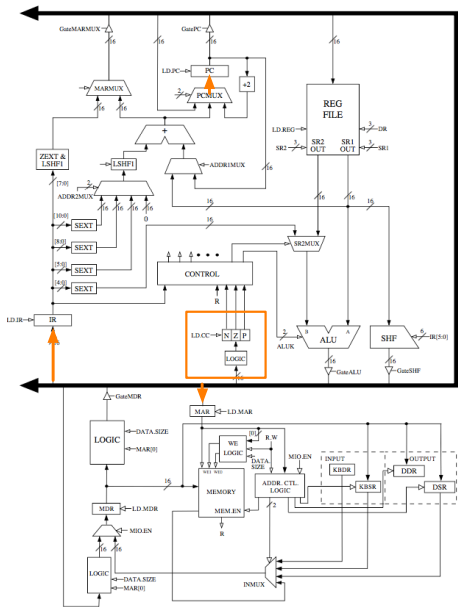
LC-3b Datapath



In lab 3-2, we write data into bus.



LC-3b Datapath



In this lab, we read data from bus:

- ▶ NEXT_LATCHES.PC
- ▶ NEXT_LATCHES.IR
- ▶ NEXT_LATCHES.MAR
- ▶ NEXT_LATCHES.N
- ▶ NEXT_LATCHES.Z
- ▶ NEXT_LATCHES.P



Overview

Introduction

Lab3-3 Assignment

Golden Results



Lab3-3 Assignment 1

- ▶ Paste your previous codes into `eval_bus_drivers()` & `drive_bus()`.
- ▶ Finish the following function.

```
671 /*
672 * datapath routine for computing all functions that need to latch
673 * values in the data path at the end of this cycle. some values
674 * require sourcing the bus; therefore, this routine has to come
675 * after drive_bus.
676 */
677 void latch_datapath_values()
678 {
679     /* LD_MDR */
680     if (GetLD_MDR(CURRENT_LATCHES.MICROINSTRUCTION))
681     {
682         if (GetMID_EN(CURRENT_LATCHES.MICROINSTRUCTION))
683         {
684             /* insert INMUX here */
685             NEXT_LATCHES.MDR = MemOut/*Low16bits((MEMORY[CURRENT_LATCHES.MAR >> 1][1] << 8) + MEMORY[CURRENT_LATCHES.MAR >> 1][0])*/;
686         }
687         else
688         {
689             NEXT_LATCHES.MDR = blockMDRLogic2(partVal(CURRENT_LATCHES.MAR, 0, 0), GetData_SIZE(CURRENT_LATCHES.MICROINSTRUCTION), BUS);
690         }
691     }
692     /* LD_BEN */
693     if (GetLD_BEN(CURRENT_LATCHES.MICROINSTRUCTION))
694     {
695         NEXT_LATCHES.BEN = (partVal(CURRENT_LATCHES.IR, 11, 11) & CURRENT_LATCHES.N) | (partVal(CURRENT_LATCHES.IR, 10, 10) & CURRENT_LATCHES.Z) | (partVal(CURRENT_LATCHES.IR, 9, 9) & CURRENT_LATCHES.P);
696     }
697     /* LD_REG */
698     if (GetLD_REG(CURRENT_LATCHES.MICROINSTRUCTION))
699     {
700         NEXT_LATCHES.REGS[blockDRMUX(GetDRMUX(CURRENT_LATCHES.MICROINSTRUCTION), partVal(CURRENT_LATCHES.IR, 11, 9), 7)] = BUS;
701     }
702     /*
703     * lab3-3 assignment in the following
704     */
705     /* LD_MAR */
706     /* LD_IR */
707     /* LD_CC */
708     /* LD_PC */
709     }
710 }
711 }
712 }
713 }
714 }
```


Lab3-3 Assignment 2: blockPCMUX

```
int blockPCMUX(int lPCMUX, int lPCAdd2, int lBUS, int  
lAdder)
```

Special treatment again! The PCMUX selection criteria is different based on your SID

Orig

lPCMUX = 0: lPCAdd2 or 1: lBUS; 2: lAdder;

SID with even ending:

lPCMUX = 0: lBUS; 1: lPCAdd2; 2: lAdder;

SID with odd ending:

lPCMUX = 0: lAdder; 1: lBUS; 2: lPCAdd2;



Overview

Introduction

Lab3-3 Assignment

Golden Results



Assignment Package

- ▶ `lc3bsim3-3.c`, `lc3bsim3.h`: codes to work on
- ▶ `libems3-3-(darwin/linux).a`: library
- ▶ `ucode3`: FSM
- ▶ `Makefile`
- ▶ `bench`: folder with benchmarks

Run the simulator:

1. `make`, then binary “`lc3bsim3-3`” is generated
2. For even ending SID: `./lc3bsim3-3 ucode3-even bench/toupper.cod`
3. For odd ending SID: `./lc3bsim3-3 ucode3-odd bench/toupper.cod`



Golden Results – case `toupper.cod`

- ▶ Please refer to [Lab 3-1 slides](#).



Golden Results – case count10.cod

- ▶ Please refer to [Lab 3-2 slides](#).



Thanks. For any question:
byu@cse.cuhk.edu.hk
wli@cse.cuhk.edu.hk

