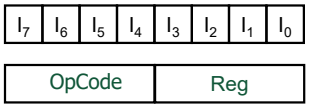
Instruction Set

ENGINEERING@SYRACUSE

9

ALU Operation Direct

ALU Operation Direct



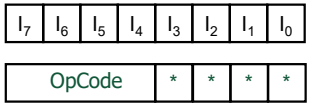
ACC = Reg OP ACC

Function Name	Function Code
Add	0001
Sub	0010
Nor	0011

Adapted from: Rabeay, J. 2003. *Digital Integrated Circuits: A Design Perspective*. Prentice Hall/Pearson.

10

ALU Operation Shift



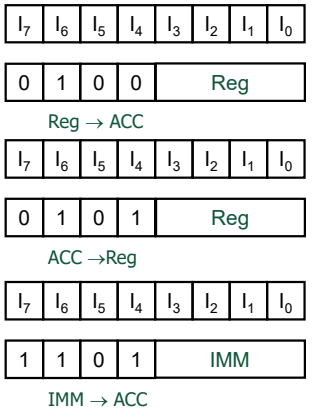
Shift ACC

Function Name	Function Code
Shift right	1100
Shift left	1011

Adapted from: Rabeay, J. 2003. *Digital Integrated Circuits: A Design Perspective*. Prentice Hall/Pearson.

11

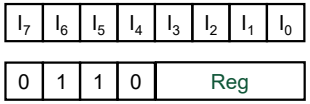
Register/ACC Load



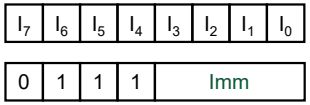
Adapted from: Rabeay, J. 2003. *Digital Integrated Circuits: A Design Perspective*. Prentice Hall/Pearson.

12

Branch Instruction



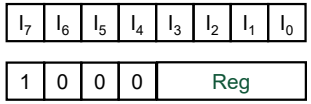
If ACC=0, Reg → PC (Jump if 0)
Else PC+1 → PC



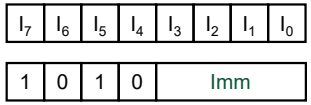
If ACC=0, Imm→ PC (Jump if 0)
Else PC+1 → PC

13

Branch Instruction



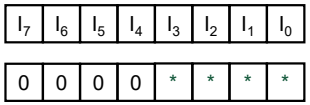
If ACC<0, Reg → PC (Jump if carry)
Else PC+1 → PC



If ACC<0, Imm→ PC (Jump if carry)
Else PC+1 → PC

14

Control Instruction



NOP

HALT

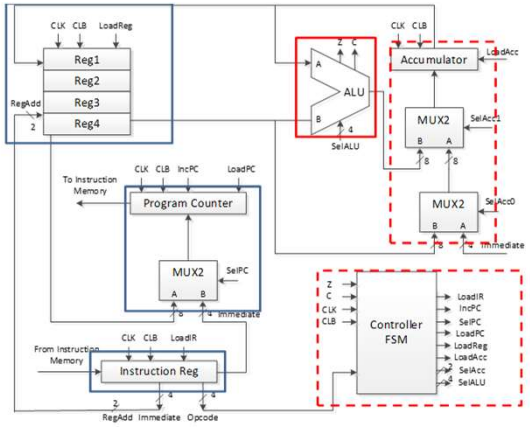
15

Architecture

ENGINEERING@SYRACUSE

16

Datapath



Adapted from: Rabeay, J. 2003. *Digital Integrated Circuits: A Design Perspective*. Prentice Hall/Pearson.

17

Controller Design

- CPI = 2
  - Cycle 1: instruction load
  - Cycle 2: Instruction decode and execution
- Assume instruction memory has asynchronous read
  - Read data (i.e. instruction) is available right after address is updated

18

Requirement

- Create Verilog files acc.v and controller.v to implement the design of accumulator and controller modules
- Create a Verilog file: processor.v to implement the top-level design. The module name must be: “processor”
- Create a test program to test all instructions, translate your test program to binary code, create a vector file the binary code of the test program and the expected output
- Create a test bench Verilog file to test your design

19

ARM Architecture

ENGINEERING@SYRACUSE

20