Ataberk ÖKLÜ

METU EEE | 2305142

Laboratory Work 3

Preliminary work

Table of Contents

[I. Datapath Design 2](#_Toc102941603)

[A. Instruction Steps 3](#_Toc102941604)

[1. DP (ADD/SUB/AND/OR/CMP) 3](#_Toc102941605)

[2. DP (MOV: LSL LSR) 3](#_Toc102941606)

[3. Memory (LDR) 3](#_Toc102941607)

[4. Memory (STR) 3](#_Toc102941608)

[II. Controller Design 6](#_Toc102941609)

[A. Black Box Diagram 6](#_Toc102941610)

[B. Changes in Base Implementation 6](#_Toc102941611)

[C. Truth Table 6](#_Toc102941612)

[1. Condition 6](#_Toc102941613)

[2. ALU Operation 6](#_Toc102941614)

[3. Operation Decoder 7](#_Toc102941615)

[D. Controller Implementation 7](#_Toc102941616)

[III. Parametrization 7](#_Toc102941617)

# Datapath Design

Diagram

Description automatically generated

## Instruction Steps

### DP (ADD/SUB/AND/OR/CMP)

* Fetch InstMem[PC]
* Increment PC by 4
* Decode Instruction -> DP + (ADD/SUB/AND/OR) + Rd + Rn + Rm
* Set AD3 = Rd and (AD1, AD2) as (Rn, Rm)
* Set Alu Source as Registers
* Set Alu Control as (ADD/SUB/AND/OR/CMP)
* Set Alu output to go register file WD3 port.
* Set Register Write 1
* Clock Rising Edge: PC and Register file is updated

### DP (MOV: LSL LSR)

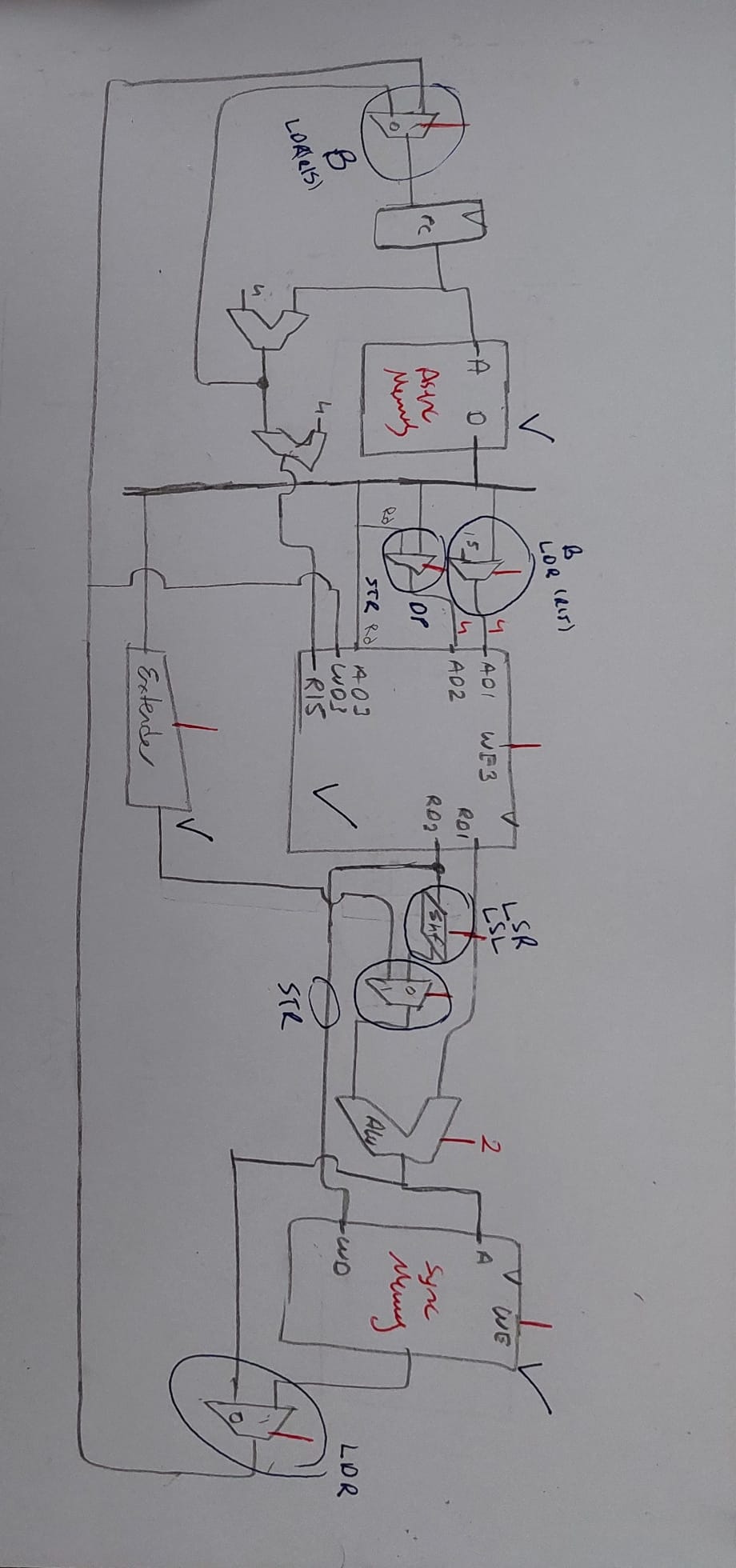
* Fetch InstMem[PC]
* Increment PC by 4
* Decode Instruction -> DP + (LSL/LSR) + Rd + Rm
* Set AD3 = Rd, AD2 = Rm
* Set Shifter Mode and Amount
* Set MemSrc as Shifter
* Forward Shifted Value to register file WD3 port by Mem2Reg.
* Set Register Write 1
* Clock Rising Edge: PC and Register file is updated

### Memory (LDR)

* Fetch InstMem[PC]
* Increment PC by 4
* Decode Instruction -> MP + (LDR) + Rd + Rn + Imm12
* Set AD3 = Rd, AD1 = Rn
* Select Immediate (ALUSrc) and Set Extender to Imm12 mode
* Set MemSrc as ALU
* Set Mem2Reg = 1, forward read data to WD3 of Register File.
* Set Register Write 1
* Clock Rising Edge: PC and Register file is updated

### Memory (STR)

* Fetch InstMem[PC]
* Increment PC by 4
* Decode Instruction -> MP + (STR) + Rd + Rn + Imm12
* Set AD1 = Rn, AD2 = Rd
* Select Immediate (ALUSrc) and Set Extender to Imm12 mode
* Set MemSrc as ALU
* Set Mem2Reg = 1 to select AD of Data Memory.
* Set WD of Data Memory as Rd
* Set Memory Write 1
* Clock Rising Edge: PC and Data Memory is updated

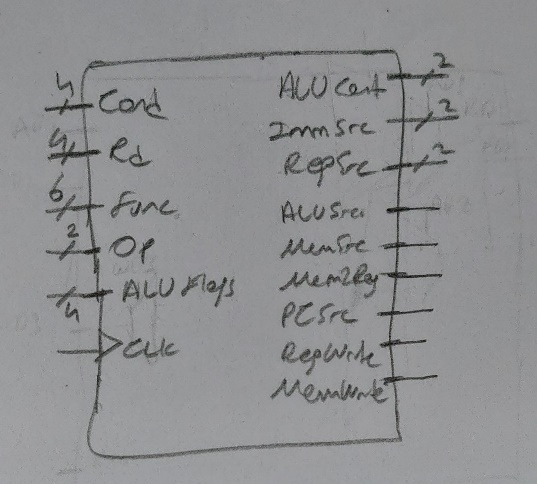


A picture containing diagram

Description automatically generated

# Controller Design

## Black Box Diagram

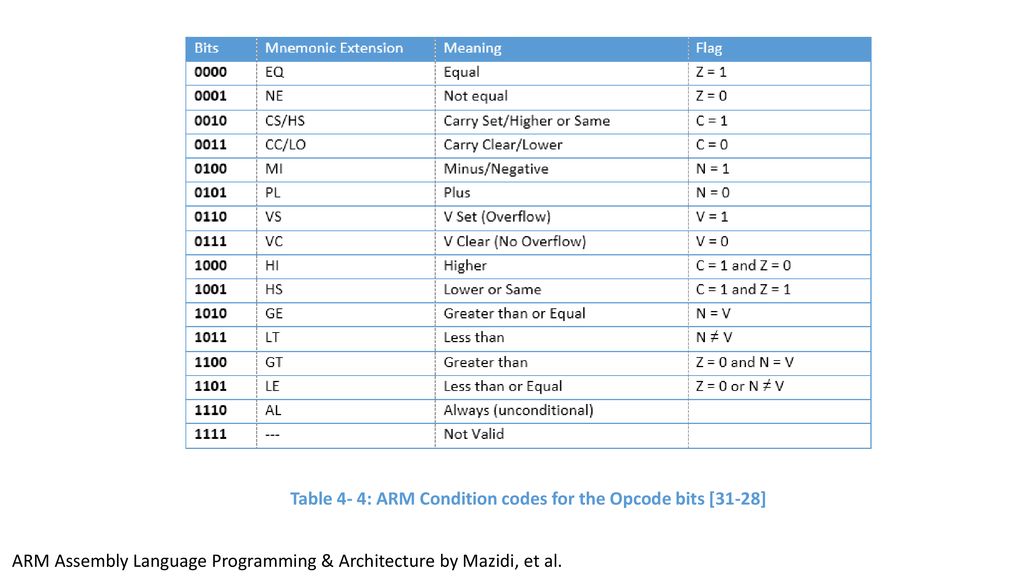


## Changes in Base Implementation

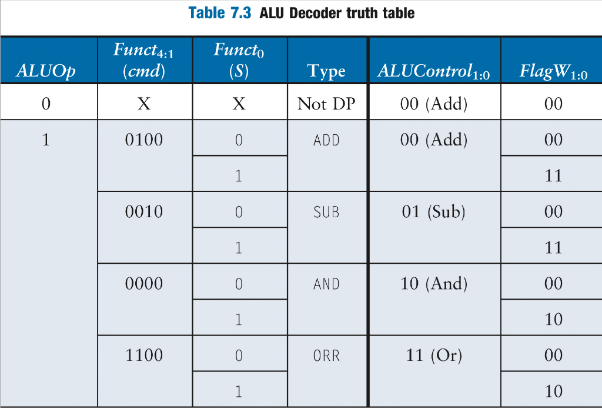
* To Implement both Data Processing and Memory Operation, Register Multiplexers are used. Their Control signals are RegSrc.
* To Implement Compare Operation, NoWrite logic is implemented in Controller in order not to update Flags
* All Other Entities are preserved from base implementation.

## Truth Table

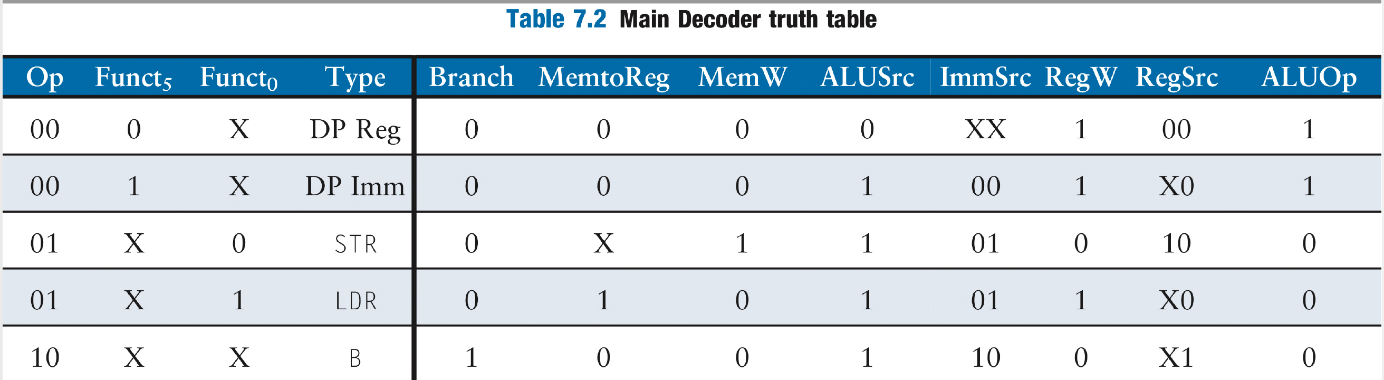
### Condition



### ALU Operation



### Operation Decoder



## Controller Implementation

Diagram, schematic

Description automatically generated

# Parametrization

* All Registers, Memories, Decoders and Muxers are implemented in the way that Data Width are parametric. Hence, we can scale each entity and overall design.
* Memory depth is configured by the help of parametrization so that data and instruction memory fits in the FPGA.
* Complications: Since the top layer (Datapath) is implemented as Board Design File, changing parameters requires manual settings of bus widths.
* This design should work various memory depths.