EE 446 LAB #3 PRELIMINARY WORK – KAAN FURKAN ALTINOK 2030021 FA1

Instructions need to be fetched with the address of PC from instruction memory. Fetched instructions are fed to control unit, register file and extend module.

PC is incremented by 4.

LDR: At ALU the data address is computed, and the data is read. The read data is fed to register file to be written.

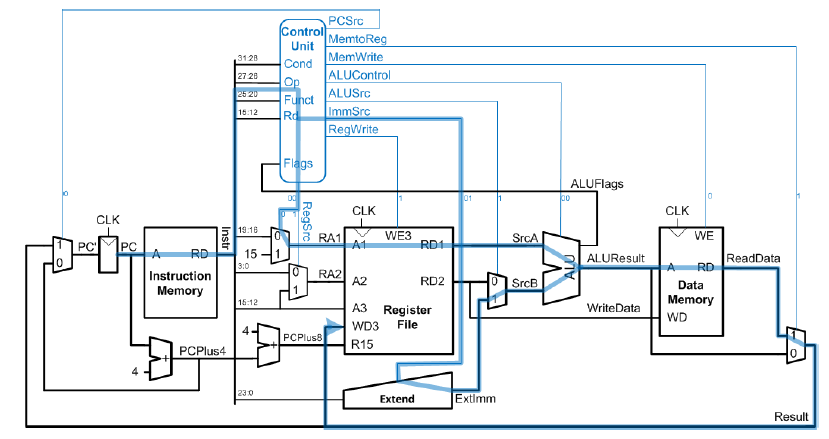


Figure 1 Load Operation

STR: Store address is computed in ALU and data to be written is coming from register files output.

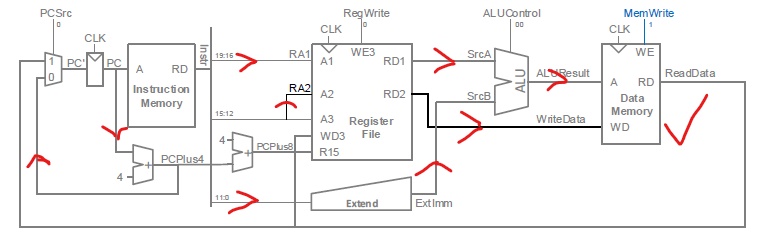


Figure 2 Store operation

Data Inst: In ALU the needed operation is computed, and the result is written back to a register in register file.

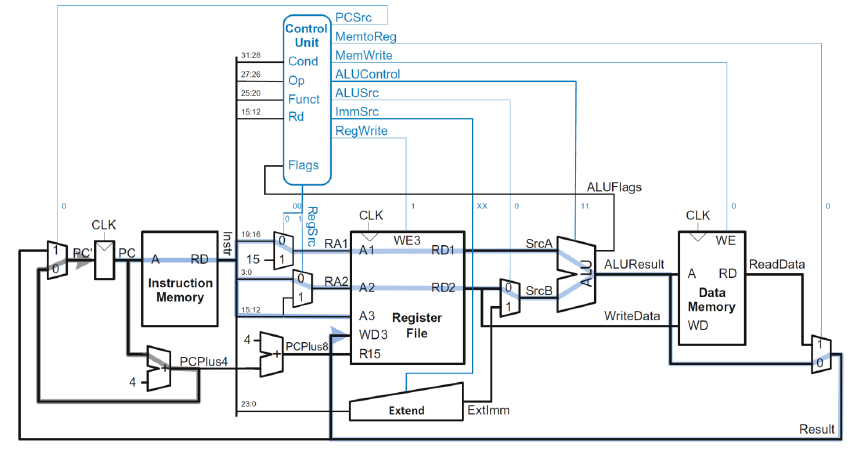


Figure 3 And, or, sub, add, cmp operations, (cmp has no writeback)

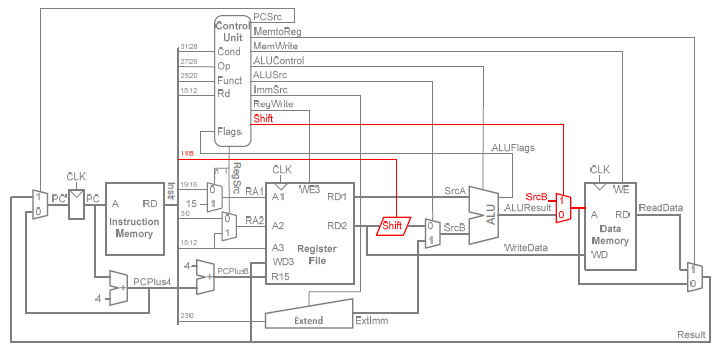
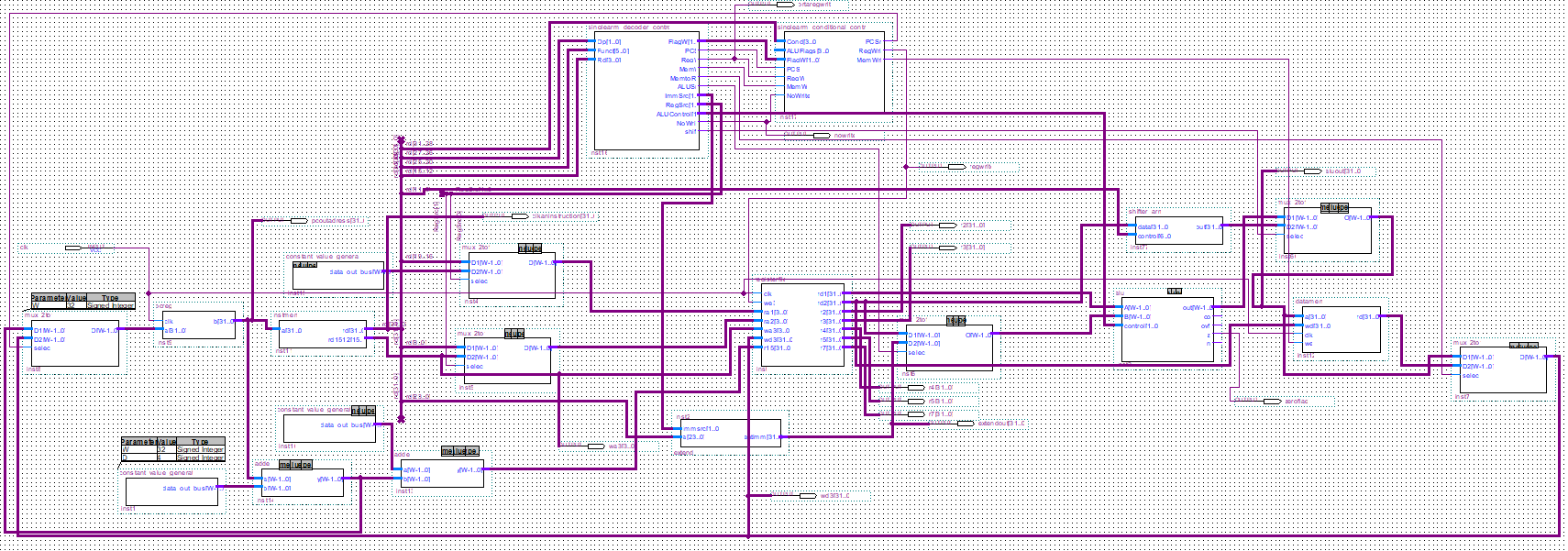
LSR,LSL: A shifter module is used. The module shifts the data by required amount and operation. The result is written back to register file.

Figure 4 Shift Operation

CMP: It is similar to subtract operation, but the result is not written to register file and the zero flag is set if the result is zero.

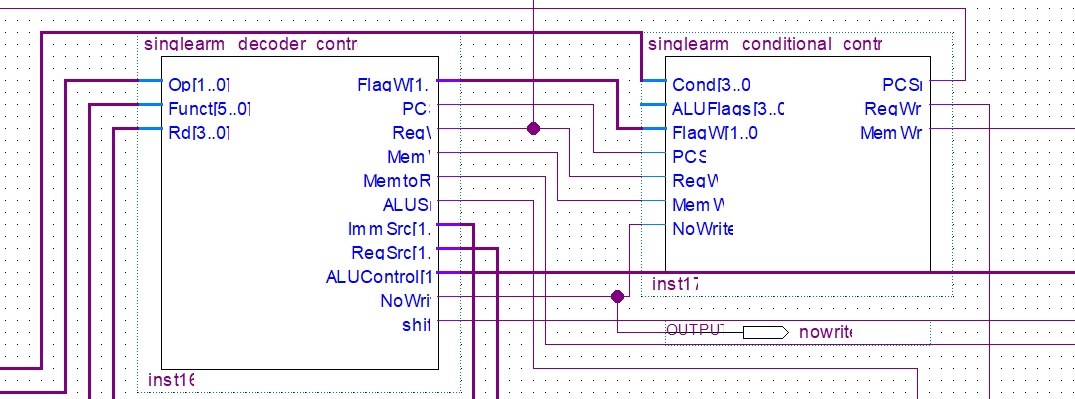
Cond, Op, Funct, Rd, Flags are my control signal inputs. They are fed by instruction bits.

I looked all of the above information from our EE445 lecture notes and implemented it on schematic editor. Below the datapath can be seen.



My controller unit consists of two blocks which are decoder and conditional namely.

As a black box one can see it in the Figure below.



For DP Reg (Op=00 Funct5=0 Funct0=X) instructions these signals need to be produced: Branch=0, MemtoReg=0, MemW=0, ALUSrc=0, RegW=1, RegSrc=00, ALUOp=1

For DP Imm (Op=00 Funct5=1 Funct0=X) instructions these signals need to be produced: Branch=0, MemtoReg=0, MemW=0, ALUSrc=1, ImmSrc=00, RegW=1, RegSrc=X0, ALUOp=1

For STR (Op=01 Funct5=X Funct0=0) instructions these signals need to be produced: Branch=0, MemtoReg=X, MemW=1, ALUSrc=1, ImmSrc=01, RegW=0, RegSrc=10, ALUOp=0

For LDR (Op=01 Funct5=X Funct0=1) instructions these signals need to be produced: Branch=0, MemtoReg=1, MemW=0, ALUSrc=1, ImmSrc=10, RegW=1, RegSrc=X0, ALUOp=0

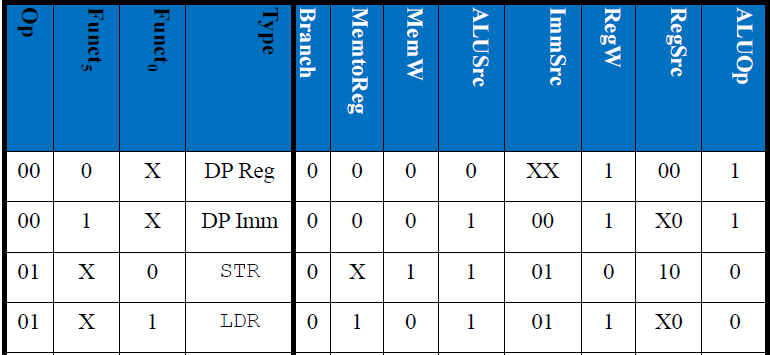
For CMP instruction additional NoWrite signal is inserted. The rest is similar to DP Reg instructions.

For Shift instruction additional Shift signal and Shifter hardware is implemented.

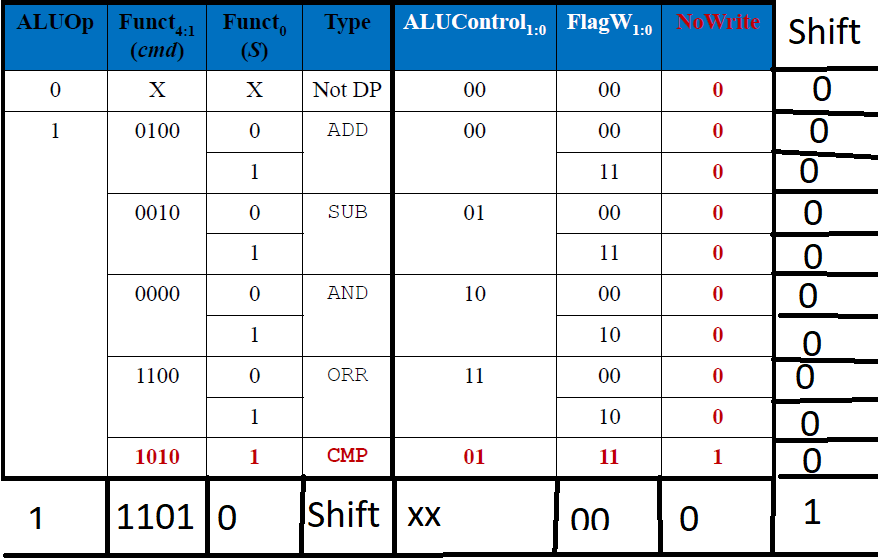
Conditional Logic is for conditional instructions and for updating ALU flags.

For DP instructions ALUControl signal is produced by ALU Decoder inside Decoder unit. Funct [4:1] bits produce proper ALUControl signals.

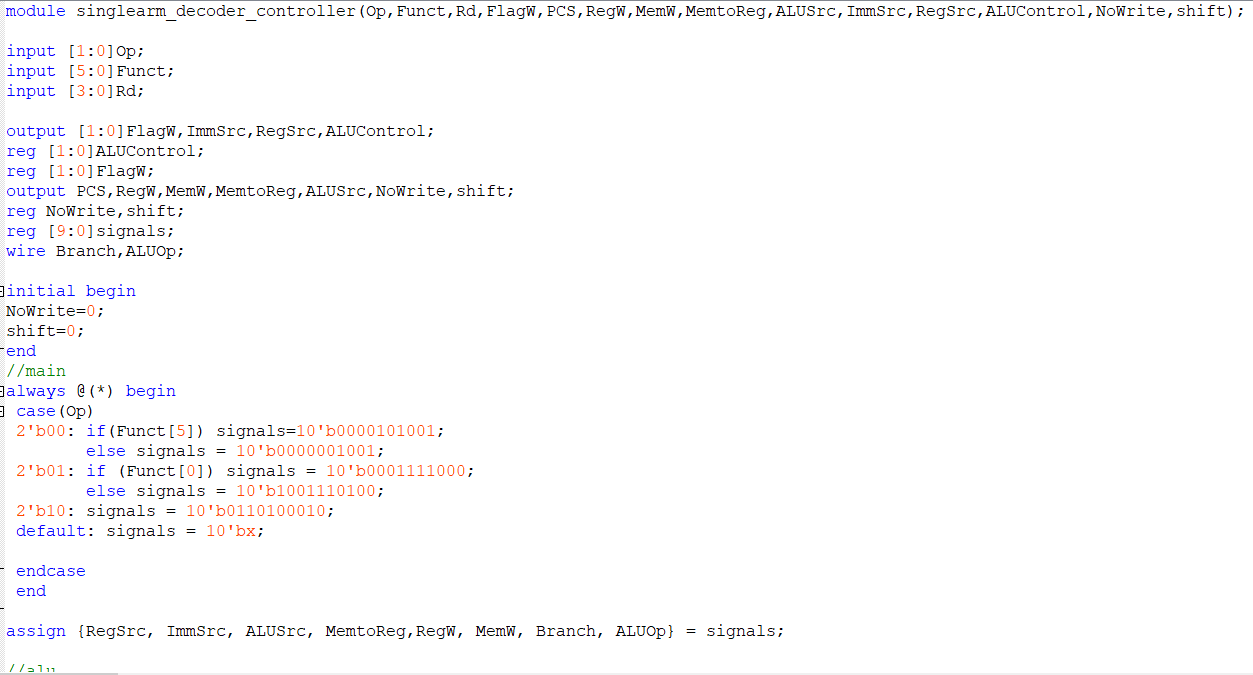
Main Decoders truth table can be seen below.

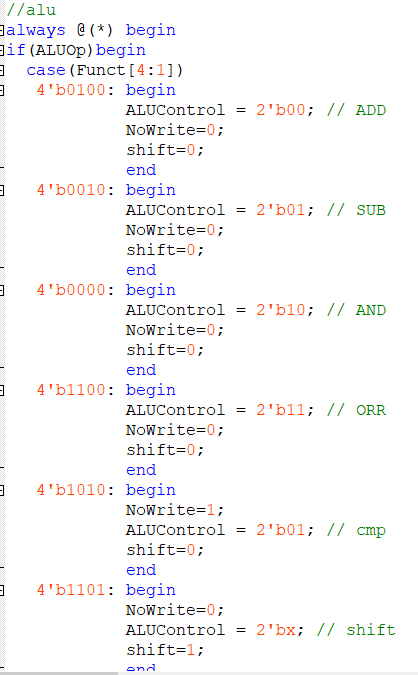


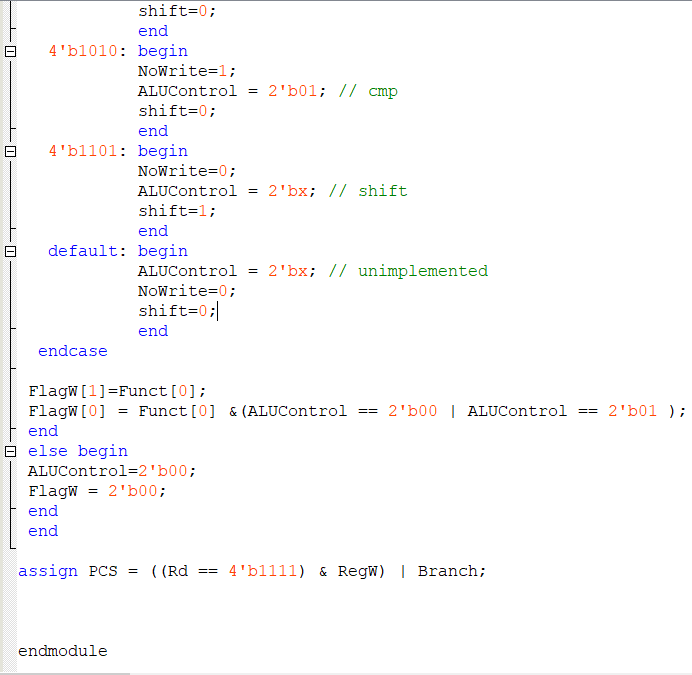
ALU Decoders truth table can be seen below.



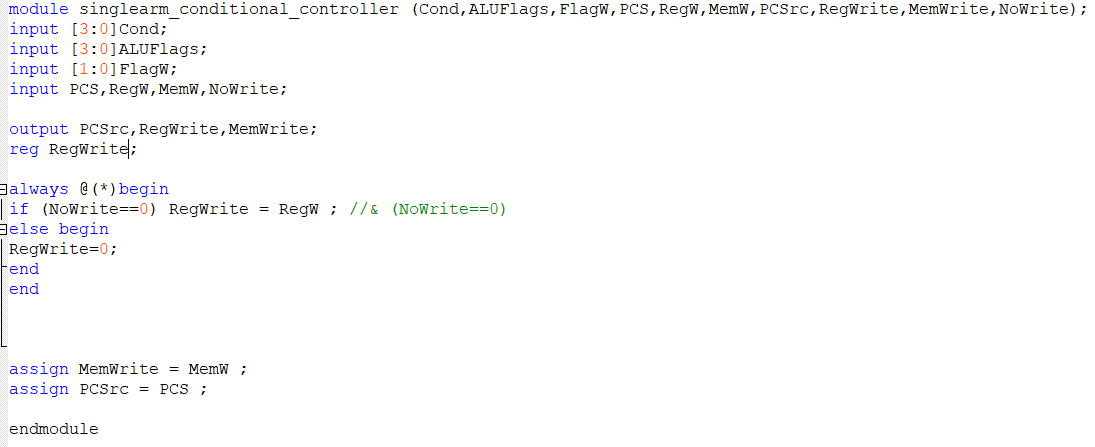
Decoder Controller code can be seen below.







Conditional Controller code can be seen below.



My single cycle arm implementation is fully working in Universal Program VWF.