13/05/2019

EE446 LABORATORY

EXPERIMENT 5

PRELIMINARY REPORT

Muttalip Caner TOL

2031466

Tuesday Afternoon

1.2. ISA Configuration

* Memory Instructions:
  + LDM Rd, [imm]
  + LDD Rd, imm
  + STD Rd, [imm]
* Arithmetic Instructions:
  + ADD Rd, Rn, Rm
  + SUB Rd, Rn
  + ADDI Rd, Rn, [Rm]
  + SUBI Rdi Rn, [Rm]
  + SUBI Rd, Rn, [Rm]
* Logic Instructions:
  + AND Rd, Rn, Rm
  + XOR Rd, Rn, Rm
  + CLR Rd
  + ORR Rd, Rn, Rm
* Shift Instructions:
  + LSL Rd, Rn
  + LSR Rd, Rn
  + ROR Rd, Rn
  + ROL Rd, Rn
  + ASR Rd, Rn
* Branch Instructions:
  + BUN [imm]
  + BLD [imm]
  + BLI [Rm]
  + BEQ [imm]
  + BNE [imm]
  + BCS [imm]
  + BCC [imm]

We have 8 registers in total which are R0, R1, R2, R3, R4, R5, LR and PC.

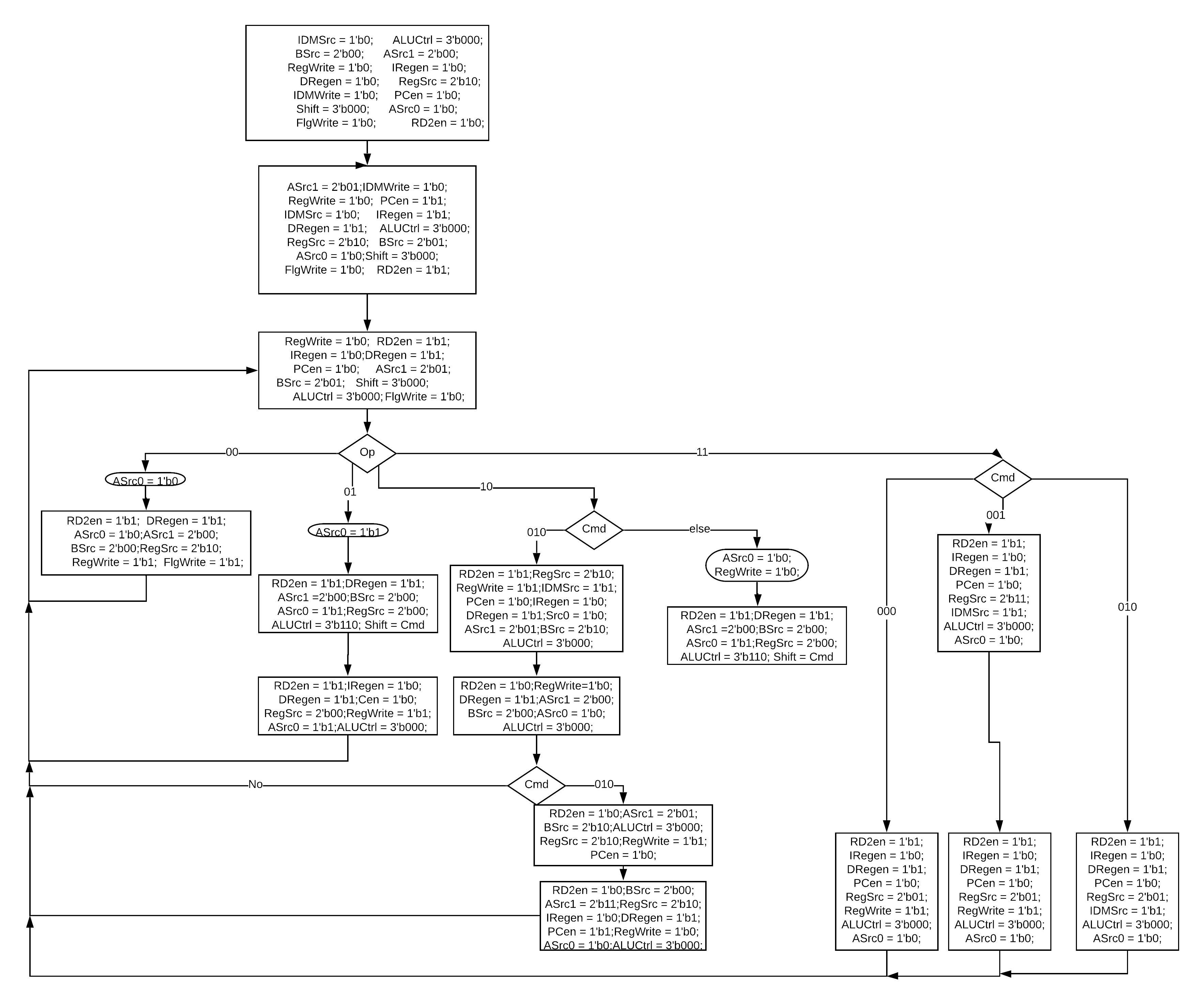
|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| ADD, SUB, XOR,CLR | Op | Cmd | Rd | Rn | Rm | 00 |
| ADDI, SUBI | Op | Cmd | Rd | Rn | Memory Address | |
| SHIFT | Op | Cmd | Rd | Rd | 00000 | |
| LDD | Op | Cmd | Rd | Data | | |
| LDM/ STD | Op | Cmd | Rd | Memory Address | | |
| Bxx | Op | Cmd | Branch Address | | | |
|  | **2 bit** | **3 bit** | **3 bit** | **3 bit** | **3 bit** | **2 bit** |
|  |  | | | **Data length: 8 bit** | | |
|  | **Total instruction length: 16 bit** | | | | | |

Arithmetic and logic instructions use Register file, ALU and IDM (Instruction / Data Memory).

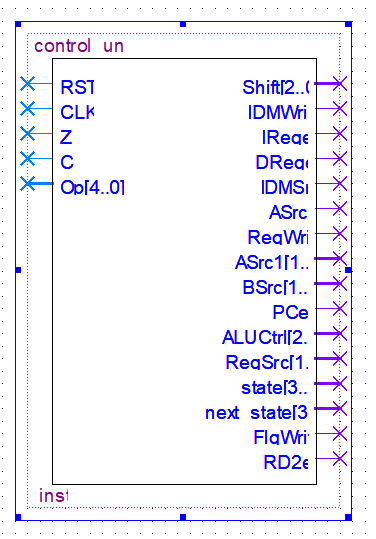
Shift instructions use Shifter, Register File and IDM.

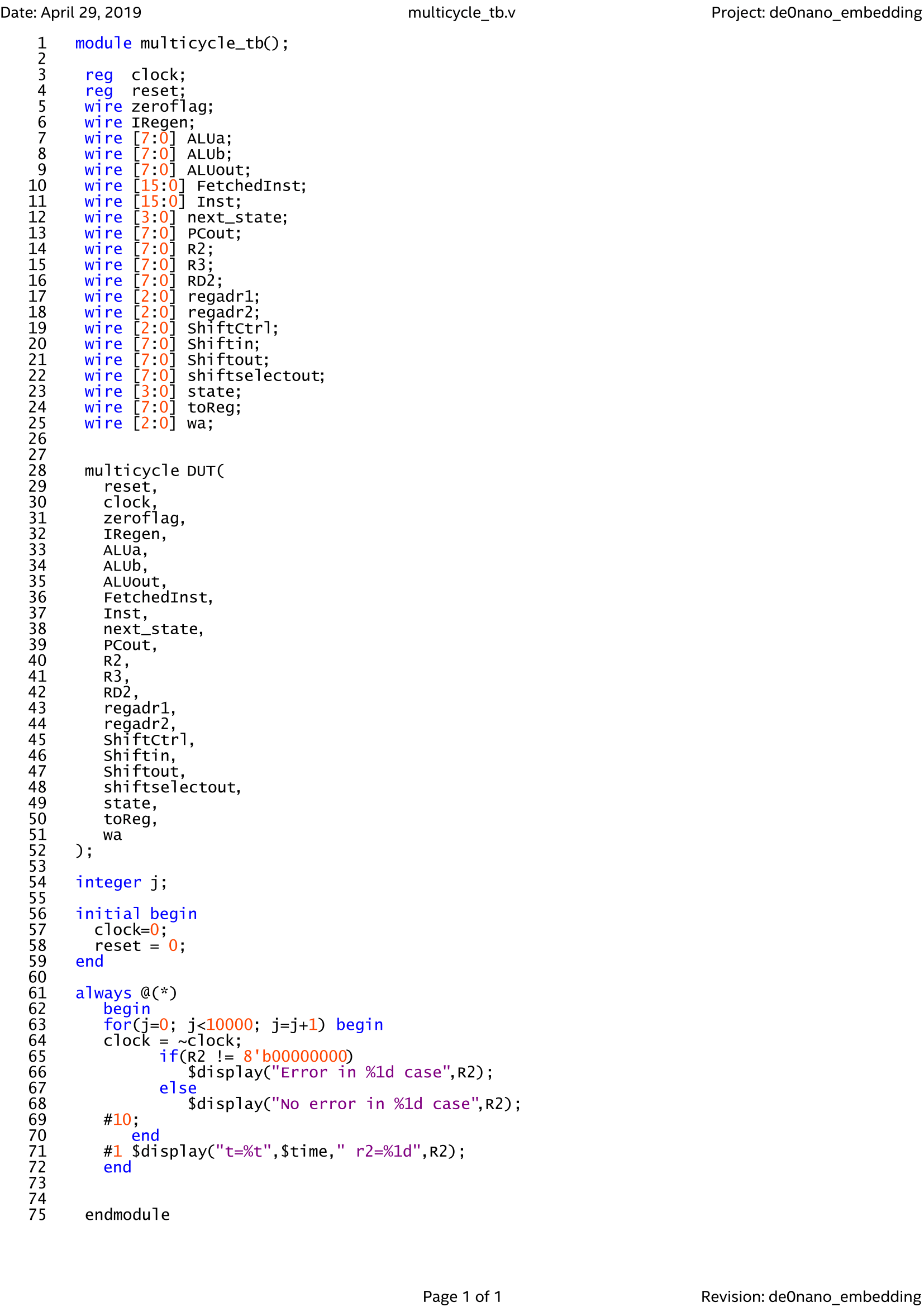
Load/Store type of instructions use Register file and IDM.

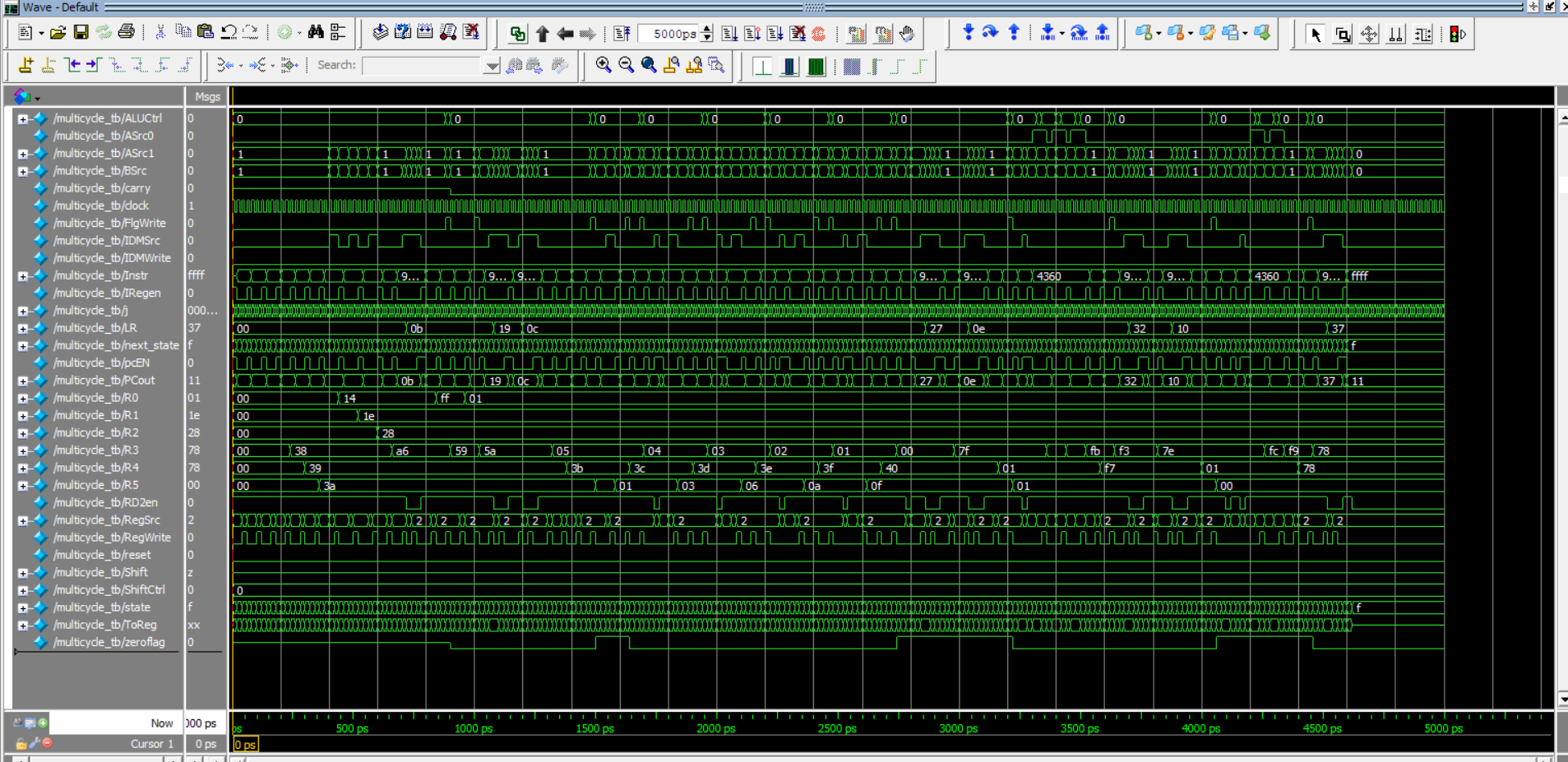
|  |  |  |  |
| --- | --- | --- | --- |
|  | Inst. | Cmd | Op |
| Arithmetic & Logic Inst. | ADD | 000 | 00 |
| SUB | 001 |
| ADDI | 010 |
| SUBI | 011 |
| AND | 100 |
| OR | 101 |
| XOR | 110 |
| CLR | 111 |
| Shift Inst. | ROL | 000 | 01 |
| ROR | 001 |
| LSL | 010 |
| ASR | 011 |
| LSR | 100 |
| Branch Inst. | BUN | 000 | 10 |
| BLD | 001 |
| BLI | 010 |
| BEQ | 011 |
| BNE | 100 |
| BCS | 101 |
| BCC | 110 |
| Memory Inst. | LDD | 000 | 11 |
| LDM | 001 |
| STD | 010 |

 ASM Chart

Control Unit



Testbench

Simulation Result