OpCode table :

|  |  |  |  |
| --- | --- | --- | --- |
| operation | regWrite | selOp | destSrc |
| load | 1 | 000 | 00 |
| mov | 1 | 000 | 01 |
| add | 1 | 000 | 10 |
| sub | 1 | 001 | 10 |
| and | 1 | 010 | 10 |
| or | 1 | 011 | 10 |
| xor | 1 | 100 | 10 |
| store | 0 | 000 | 11 |

|  |  |
| --- | --- |
| operation | opcode |
| load | 000 |
| mov | 001 |
| add | 010 |
| sub | 011 |
| and | 100 |
| or | 101 |
| xor | 110 |
| store | 111 |

Control Signals:

ALU operations:

|  |  |
| --- | --- |
| operation | selOp |
| add | 000 |
| sub | 001 |
| and | 010 |
| or | 011 |
| xor | 100 |

Mux select:

|  |  |
| --- | --- |
| operation | sel |
| dataIn | 00 |
| dataA | 01 |
| ALUout | 10 |
| dataB | 11 |

Outlines:

* Eight ( 8 bits registers)
* Two (eight bits ports) : dataIn, dataOut.
* 8 operations
* Function format(9-bits):
  + 3-bits -> opcode
  + 3-bits -> distanation address
  + 3-bits -> source address
* Register file consist of 8x8bits registers
* DataA and dataB : 8 bits output data for the .. of the two registers
* dataW : 8 bits input for writing to a register when regWrite is 1
* registers are selected by:
  + Ra :…
  + Rb:…
  + Rw:…
* Clock input :
  + Clock is only for write
  + During read the register file acts as a combinational circtuit