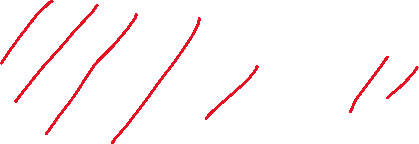
|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| NOT | 1 | 1 | 0 | 0 | 0 |  |
| INC | 1 | 1 | 0 | 0 | 1 |  |
| DEC | 1 | 1 | 0 | 1 | 0 |  |
| MOV | 1 | 1 | 0 | 1 | 1 |  |
| ADD | 1 | 1 | 1 | 0 | 0 |  |
| IADD | 1 | 1 | 1 | 0 | 0 | 1 |
| SUB | 1 | 1 | 1 | 0 | 1 |  |
| AND | 1 | 1 | 1 | 1 | 0 |  |
| OR | 1 | 1 | 1 | 1 | 1 |  |
| PUSH | 0 | 1 | 0 | 0 | 0 |  |
| POP | 0 | 1 | 0 | 0 | 1 |  |
| LDM | 0 | 1 | 0 | 1 | 0 | 1 |
| LDD | 0 | 1 | 0 | 1 | 1 |  |
| STD | 0 | 1 | 1 | 0 | 0 |  |
| JZ | 1 | 0 | 0 | 0 | 0 |  |
| JC | 1 | 0 | 0 | 0 | 1 |  |
| JMP | 1 | 0 | 0 | 1 | 0 |  |
| CALL | 1 | 0 | 0 | 1 | 1 |  |
| RET | 1 | 0 | 1 | 0 | 0 |  |
| RTI | 1 | 0 | 1 | 0 | 1 |  |
| NOP | 0 | 0 | 0 | 0 | 0 |  |
| SETC | 0 | 0 | 0 | 0 | 1 |  |
| CLRC | 0 | 0 | 0 | 1 | 0 |  |
| IN | 0 | 0 | 0 | 1 | 1 |  |
| OUT | 0 | 0 | 1 | 0 | 0 |  |

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | **RegWrite** | **ALUSrc** | **PCSrc** | **MemRead (used by memory)** | **MemWrite**  **(used by memory)** | **MemToReg**  **(used by WB)** | **IN**  **(used by WB)** | **OUT**  **(used by WB)** | **SETC** | **CLRC** | **SP+**  **(used by memory)** | **SP-**  **(used by memory)** |
| NOT | 1 |  |  |  |  |  |  |  |  |  |  |  |
| INC |  |  |  |  |  |  |  |  |  |  |  |
| DEC |  |  |  |  |  |  |  |  |  |  |  |
| MOV |  |  |  |  |  |  |  |  |  |  |  |
| ADD |  |  |  |  |  |  |  |  |  |  |  |
| IADD | 1 |  |  |  |  |  |  |  |  |  |  |
| SUB |  |  |  |  |  |  |  |  |  |  |  |
| AND |  |  |  |  |  |  |  |  |  |  |  |
| OR |  |  |  |  |  |  |  |  |  |  |  |
| PUSH |  |  |  |  | 1 |  |  |  |  |  |  | 1 |
| POP | 1 |  |  | 1 |  | 1 |  |  |  |  | 1 |  |
| LDM | 1 | 1 |  |  |  |  |  |  |  |  |  |  |
| LDD | 1 |  |  | 1 |  | 1 |  |  |  |  |  |  |
| STD |  |  |  |  | 1 |  |  |  |  |  |  |  |
| JZ |  |  | 1 |  |  |  |  |  |  |  |  |  |
| JC |  |  |  |  |  |  |  |  |  |  |  |
| JMP |  |  |  |  |  |  |  |  |  |  |  |
| CALL |  |  |  | 1 |  |  |  |  |  |  | 1 |
| RET |  |  | 1 |  |  |  |  |  |  | 1 |  |
| RTI |  |  | 1 |  |  |  |  |  |  | 1 |  |
| NOP |  |  |  |  |  |  |  |  |  |  |  |  |
| SETC |  |  |  |  |  |  |  |  | 1 |  |  |  |
| CLRC |  |  |  |  |  |  |  |  |  | 1 |  |  |
| IN | 1 |  |  |  |  |  | 1 |  |  |  |  |  |
| OUT |  |  |  |  |  |  |  | 1 |  |  |  |  |

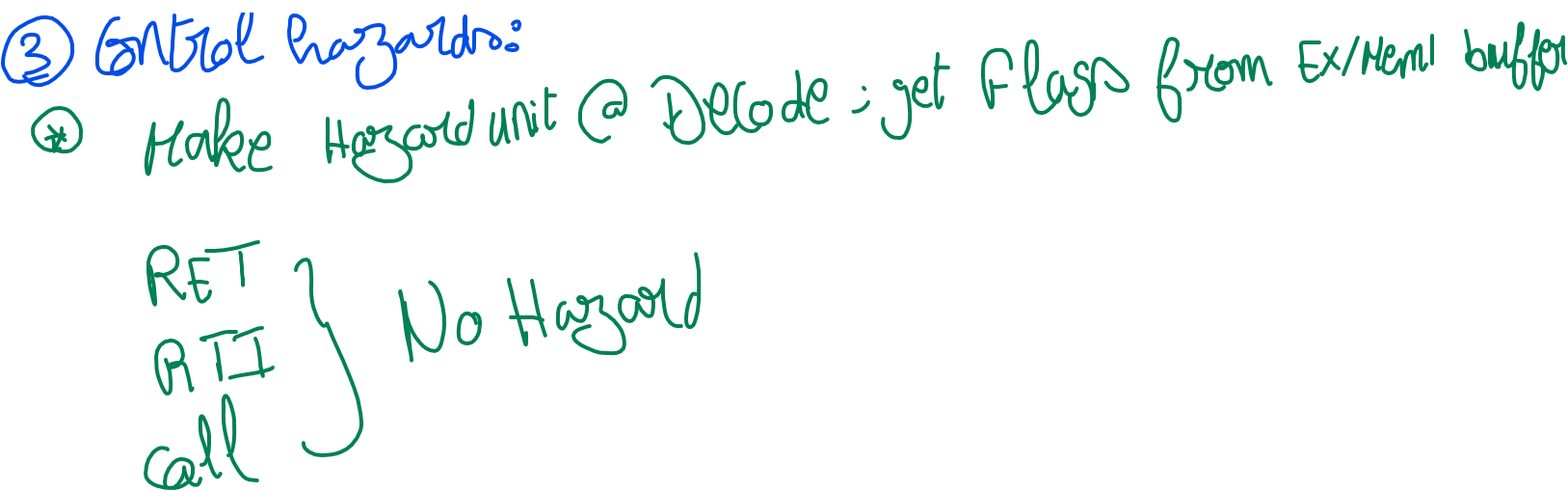
Text

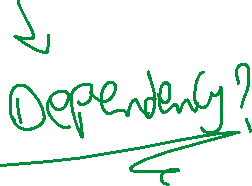
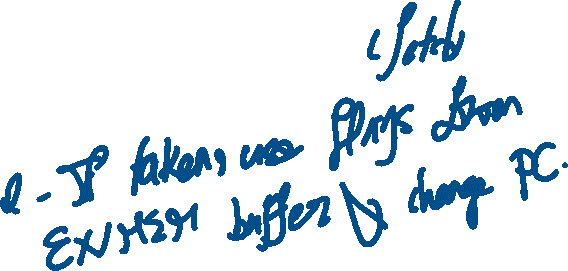
Description automatically generated



Text

Description automatically generated





|  |  |  |
| --- | --- | --- |
| **Register** | **Inputs** | **Outputs** |
| IF/ID |  |  |
| ID/EX | * RS1 data * RS2 data * Instruction as a whole [will split inside] * Immediate * All control signals * PC | * RS1 data * RS2 data * Opcode * Immediate * isImmediate * Rd * Rs1 * Rs2 * PC * All control signals |
| EX/MEM1 | * Alu output * Flag register [or not] (might keep inside execution stage for jumps) * Rs2Data [for memory write] * Rs1Data [for out – used at write back] * Memory signals * PC | * Alu output * Rs2 * Memory signals * PC |
| MEM1/MEM2 |  |  |
| MEM2/WB |  |  |