ISA

Programmer Visible State:

| Program Registers | |
| --- | --- |
| %eax | %esp |
| %ecx | %ebp |
| %edx | %esi |
| %ebx | %edi |

| Condition Codes | | |
| --- | --- | --- |
| Symbol | Name | Use |
| ZF | Zero Flag | Return 1 if 0 |
| SF | Sign Flag | Return 1 if negative |
| OF | Overflow Flag | Return 1 if overflow occurs |

| Name | Code | | | | | Use |
| --- | --- | --- | --- | --- | --- | --- |
| Icode  4 bits | Ifun  4 bits | rA  4 bits | rB  4 bits | valC  (4 bytes) |
| halt | 0 | 0 | - | - | - | Halts program |
| nop | 1 | 0 | - | - | - | No operation |
| crmovXX rA, rB | 2 | X | rA | rB | - | Conditional move statement from rA -> rB depending on function code |
| irmovl V, rB | 3 | 0 | F | rB | V | Moves immediate value from V -> rB |
| rmmovl rA,D(rB) | 4 | 0 | rA | rB | D(rB) | Writes value from register to memory rA -> D(rB) |
| mrmovl D(rB), rA | 5 | 0 | rA | rB | D(rB) | Writes value from memory to register D(rB) -> rA |
| OP rA,rB | 6 | X | rA | rB | - | Does an operation with the values in rA and rB depending on the function code used with it |
| jXX Dest | 7 | X | - | - | Dest | Jumps to memory address depending on what condition the function code gives |

Operation Codes:

| Operations | Icode  4 bits | Ifun  4 bits | Use |
| --- | --- | --- | --- |
| add | 6 | 0 | Adds values in registers rA and rB and stores it in register rB |
| sub | 6 | 1 | Multiplies values in registers rA and rB and stores it in register rB |
| AND | 6 | 2 | Subtracts value in rA from rB and stores in rB |
| XOR | 6 | 3 | Divides value in rA from rB and stores in rB |
| mult | 6 | 4 | multiplies value in rA from rB and stores in rB |
| div | 6 | 5 | Divides value in rA by the value in rB and stores in rB |

Jx Condition Codes

| Jump | Icode  4 bits | Ifun  4 bits | Name |
| --- | --- | --- | --- |
| jmp | 7 | 0 | Jumps to memory address unconditionally |
| jle | 7 | 1 | Jumps to memory address if last operation is less than or equal to |
| jl | 7 | 2 | Jumps to memory address if last operation is less than |
| je | 7 | 3 | Jumps to memory address if last operation is zero |
| jne | 7 | 4 | Jumps to memory address if last operation is not zero |
| jge | 7 | 5 | Jumps to memory address if last operation is greater than or equal to |
| jg | 7 | 6 | Jumps to memory address if last operation is less than or equal to |

CMx Condition Codes

| Jump | Icode  4 bits | Ifun  4 bits | Name |
| --- | --- | --- | --- |
| rrmov | 7 | 0 | Moves rA to rB unconditionally |
| cmovle | 7 | 1 | Moves rA to rB if last operation is less than or equal to |
| cmovl | 7 | 2 | Moves rA to rB if last operation is less than |
| cmove | 7 | 3 | Moves rA to rB if last operation is zero |
| cmovne | 7 | 4 | Moves rA to rB if last operation is not zero |
| cmovge | 7 | 5 | Moves rA to rB if last operation is greater than or equal to |
| cmovg | 7 | 6 | Moves rA to rB if last operation is less than or equal to |

Program Register Identifiers:

| Number | Register | Use |
| --- | --- | --- |
| 0 | %eax | Return register |
| 1 | %ecx | Argument 1 |
| 2 | %edx | Argument 2 |
| 3 | %ebx | Argument 3 |
| 4 | %esp | Argument 4 |
| 5 | %ebp | Argument 5 |
| 6 | %esi | Argument 6 |
| 7 | %edi | Argument 7 |