**Transformation Tables**

**Group 25 - Noah Mustoe, Yuan Wang, Josh Pop**

| **Instructions** | **Fetch** |
| --- | --- |
| rrmovq rA, rB | icode:ifun ← M1[PC]  rA:rB ← M1[PC + 1] |
| irmovq V, rB | icode:ifun ← M1[PC]  rA:rB ← M1[PC + 1]  valC ← M8[PC + 2] |
| rmmovq rA, D(rB) | icode:ifun ← M1[PC]  rA:rB ← M1[PC + 1]  valC ← M8[PC + 2]  valP ← PC + 10 |
| mrmovq D(rB), rA | icode:ifun ← M1[PC]  rA:rB ← M1[PC + 1]  valC ← M8[PC + 2]  valP ← PC + 10 |
| OPq rA, rB | icode:ifun ← M1[PC]  rA:rB ← M1[PC + 1] |
| jXX Dest | icode:ifun ← M1[PC]  valC ← M8[PC + 1]  valP ← PC + 9 |
| call Dest | icode:ifun ← M1[PC]  valC ← M8[PC + 1]  valP ← PC + 9 |
| ret | icode:ifun ← M1[PC]  valP ← PC + 1 |
| pushq rA | icode:ifun ← M1[PC]  rA:rB ← M1[PC + 1]  valP ← PC + 2 |
| popq rA | icode:ifun ← M1[PC]  rA:rB ← M1[PC + 1]  valP ← PC + 2 |

| **Instructions** | **Decode** |
| --- | --- |
| rrmovq rA, rB | valA ← R[rA]  valB ← R[rB] |
| irmovq V, rB | N/A |
| rmmovq rA, D(rB) | valA ← R[rA]  valB ← R[rB] |
| mrmovq D(rB), rA | valB ← R[rB] |
| OPq rA, rB | valA ← R[rA]  valB ← R[rB] |
| jXX Dest | N/A |
| call Dest | valA ← R[%rsp]  valB ← R[%rsp] |
| ret | valA ← R[%rsp]  valB ← R[%rsp] |
| pushq rA | valA ← R[rA]  valB ← R[%rsp] |
| popq rA | valA ← R[%rsp]  valB ← R[%rsp] |

| **Instructions** | **Execute** |
| --- | --- |
| rrmovq rA, rB | valE ← valB + valC |
| irmovq V, rB | valE ← 0 + valC |
| rmmovq rA, D(rB) | valE ← valB + valC |
| mrmovq D(rB), rA | valE ← valB + valC |
| OPq rA, rB | valE ← valA OP valB  Set CC |
| jXX Dest | Cnd ← Cond(CC, ifun) |
| call Dest | valE ← valB + (-8) |
| ret | valE ← valB + 8 |
| pushq rA | valE ← valB + (-8) |
| popq rA | valE ← valB + 8 |

| **Instructions** | **Memory** |
| --- | --- |
| rrmovq rA, rB | N/A |
| irmovq V, rB | N/A |
| rmmovq rA, D(rB) | M8[valE] ← valA |
| mrmovq D(rB), rA | valM ← M8[valE] |
| OPq rA, rB | N/A |
| jXX Dest | N/A |
| call Dest | M8[valE] ← valP |
| ret | valM ← M8[valA] |
| pushq rA | M8[valE] ← valA |
| popq rA | valM ← M8[valA] |

| **Instructions** | **Write back** |
| --- | --- |
| rrmovq rA, rB | R[rB] ← valE |
| irmovq V, rB | R[rB] ← valE |
| rmmovq rA, D(rB) | N/A |
| mrmovq D(rB), rA | R[rA] ← valM |
| OPq rA, rB | R[rB] ← valE |
| jXX Dest | N/A |
| call Dest | R[%rsp] ← valE |
| ret | R[%rsp] ← valE |
| pushq rA | R[%rsp] ← valE |
| popq rA | R[%rsp] ← valE  R[rA] ← valM |

| **Instructions** | **PC update** |
| --- | --- |
| rrmovq rA, rB | PC ← valP |
| irmovq V, rB | PC ← valP |
| rmmovq rA, D(rB) | PC ← valP |
| mrmovq D(rB), rA | PC ← valP |
| OPq rA, rB | PC ← valP |
| jXX Dest | PC ← Cnd ? valC : valP |
| call Dest | PC ← valC |
| ret | PC ← valM |
| pushq rA | PC ← valP |
| popq rA | PC ← valP |