

### **ADD (0001)** Instruction Cycle

|      |    |    |    |    |    |   |     |   |   |   |    |   |     |   |   |
|------|----|----|----|----|----|---|-----|---|---|---|----|---|-----|---|---|
| 15   | 14 | 13 | 12 | 11 | 10 | 9 | 8   | 7 | 6 | 5 | 4  | 3 | 2   | 1 | 0 |
| 0001 |    |    |    | DR |    |   | SR1 |   |   | 0 | 00 |   | SR2 |   |   |

|      |    |    |    |    |    |   |     |   |   |   |      |   |   |   |   |
|------|----|----|----|----|----|---|-----|---|---|---|------|---|---|---|---|
| 15   | 14 | 13 | 12 | 11 | 10 | 9 | 8   | 7 | 6 | 5 | 4    | 3 | 2 | 1 | 0 |
| 0001 |    |    |    | DR |    |   | SR1 |   |   | 1 | imm5 |   |   |   |   |

#### Instruction Fetch

**MAR**  $\leftarrow$  **PC**

**PC**  $\leftarrow$  **PC** + 1

**MDR**  $\leftarrow$  **memory**[**MAR**]

**IR**  $\leftarrow$  **MDR**

#### Decode

**IR**[15:12] decoded

#### Evaluate Address

n/a

#### Operand Fetch

**ALU**<sub>INA</sub>  $\leftarrow$  **R**<sub>IR[8:6]</sub>

**ALU**<sub>INB</sub>  $\leftarrow$  (**IR**[5] ? **sign\_ext**<sub>16</sub>(**IR**[4:0]) : **R**<sub>IR[2:0]</sub>)

#### Execute

**ALU**<sub>OUT</sub>  $\leftarrow$  **ALU**<sub>INA</sub> + **ALU**<sub>INB</sub>

#### Store Result

**R**<sub>IR[11:9]</sub>  $\leftarrow$  **ALU**<sub>OUT</sub>