# Instructions that affect Flag Settings

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Instruction** | **Flags** | | | **Instruction** | **Flags** | | |
| **C** | **Z** | **OV** | **C** | **Z** | **OV** |
| ADD | X | X | X | NDU |  | X |  |
| ADC | X | X | X | NDC |  | X |  |
| ADZ | X | X | X | NDZ |  | X |  |
| ADO | X | X | X | NDO |  | X |  |
| ADI | X | X | X | LW |  | X |  |

# Instruction Set and Addressing Modes

|  |  |
| --- | --- |
| Rn | Registers R6-R0 of the Register Bank |
| data 6 | Signed 6-bit constant included in instruction |
| data 9 | Signed 9-bit constant included in instruction |
| rel 6 | Signed (two’s complement) 6-bit offset byte. Used by BEQ. Range is -32 to +31 short words relative to the present instruction |
| rel 9 | Signed (two’s complement) 9-bit offset byte. Used by JAL. Range is -256 to +255 short words relative to the present instruction |
| reg | 8-bits representing registers R7-R0. Used by LM and SM |
| R7 | Program Counter |
| R | Registers R7-R0 of the Register Bank |

# Instruction Encoding

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| ADD: | 00\_00 | RA | RB | RC | 0 | 00 |
| ADC: | 00\_00 | RA | RB | RC | 0 | 10 |
| ADZ: | 00\_00 | RA | RB | RC | 0 | 01 |
| ADO: | 00\_00 | RA | RB | RC | 0 | 11 |
| ADI: | 00\_01 | RA | RB | data 6 | | |
| NDU: | 00\_10 | RA | RB | RC | 0 | 00 |
| NDC: | 00\_10 | RA | RB | RC | 0 | 10 |
| NDZ: | 00\_10 | RA | RB | RC | 0 | 01 |
| NDO: | 00\_10 | RA | RB | RC | 0 | 11 |
| LHI: | 00\_11 | RA | data 9 | | | |
| LW: | 01\_00 | RA | RB | rel 6 | | |
| SW: | 01\_01 | RA | RB | rel 6 | | |
| LM: | 01\_10 | RA | reg | | | |
| SM: | 01\_11 | RA | reg | | | |
| LLI: | 10\_11 | RA | data 9 | | | |
| BEQ: | 11\_00 | RA | RB | rel 6 | | |
| JAL: | 10\_00 | RA | rel 9 | | | |
| JLR: | 10\_01 | RA | RB | 000\_000 | | |

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Mnemonic | | Flags | | | Oscillator Period |  | Mnemonic | | Flags/  Condition | Oscillator Period |
| C | Z | OV |
| ADD | Rn, R, R | X | X | X | 5 | ADI | Rn, R, data6 | X | 5 |
| ADD | R7, R, R | X | X | X | 4 | ADI | R7, R, data6 | X | 4 |
| ADC | Rn, R, R | 0 | X | X | 5 | LHI | Rn, data9 | X | 3 |
| ADC | R7, R, R | 0 | X | X | 4 | LHI | R7, data9 | X | 2 |
| ADC | Rn, R, R | 1 | X | X | 3 | LLI | Rn, data9 | X | 3 |
| ADZ | Rn, R, R | X | 0 | X | 5 | LLI | R7, data9 | X | 2 |
| ADZ | R7, R, R | X | 0 | X | 4 | LW | R, R, rel6 | X | 5 |
| ADZ | Rn, R, R | X | 1 | X | 3 | SW | R, R, rel6 | X | 4 |
| ADO | Rn, R, R | X | X | 0 | 5 | LM | R, reg | X | 3+2\*reg\_count |
| ADO | R7, R, R | X | X | 0 | 4 | SM | R, reg | X | 3+2\*reg\_count |
| ADO | Rn, R, R | X | X | 1 | 3 | JAL | Rn, rel9 | X | 3 |
| NDU | Rn, R, R | X | X | X | 5 | JAL | R7, rel9 | X | 2 |
| NDU | Rn, R, R | X | X | X | 4 | JLR | Rn, R | X | 3 |
| NDC | Rn, R, R | 0 | X | X | 5 | JLR | R7, R | X | 2 |
| NDC | R7, R, R | 0 | X | X | 4 | BEQ |  | Not Equal | 3 |
| NDC | Rn, R, R | 1 | X | X | 3 | BEQ |  | Equal | 4 |
| NDZ | Rn, R, R | X | 0 | X | 5 |  |  |  |  |
| NDZ | R7, R, R | X | 0 | X | 4 |  |  |  |  |
| NDZ | Rn, R, R | X | 1 | X | 3 |  |  |  |  |
| NDO | Rn, R, R | X | X | 0 | 5 |  |  |  |  |
| NDO | R7, R, R | X | X | 0 | 4 |  |  |  |  |
| NDO | Rn, R, R | X | X | 1 | 3 |  |  |  |  |

# Custom Instructions

* ADO: ADD if the overflow flag is set. The rest of the instruction format is the same as the other conditional execution instructions
* NDO: NAND if the overflow flag is set. The rest of the instruction format is the same as the other conditional execution instructions
* LLI: Load lower immediate. Load the immediate 9 bits into the register as mentioned in the instruction without sign extension by padding zeros in the upper 7 bits.