ARM Reference Card for CSE 30 Fall 2020 v2

**Arithmetic Instructions**

|  |  |  |
| --- | --- | --- |
| ADD | ADD R1, R2, R3 | R[1] = R[2] + R[3] |
| SUB | SUB R1, R2, R3 | R[1] = R[2] - R[3] |
| MUL | MUL R1, R2, R3 | R[1] = R[2] \* R[3]  R1 gets only bottom 32 bits |
| [U/S]MULL | [U/S]MULL R0, R1, R2, R3 | R[1] = R[2]\*R[3][b’63-32]  R[0] = R[2]\*R[3][b’31-0] |
| MOV | MOV R0, R1 | R[0] = R[1] |
| MVN | MOVN R0, R1 | R[0] = !R[1]  ! is the 1s complement |

**Bitwise and Shift Instructions**

|  |  |  |
| --- | --- | --- |
| AND | AND R1, R2, R3 | R[1] = R[2] & R[3] |
| ORR | ORR R1, R2, R3 | R[1] = R[2] | R[3] |
| EOR | EOR R1, R2, R3 | R[1] = R[2] ^ R[3] |
| BIC | BIC R1, R2, R3 | R[1] = R[2] & !R[3] |
| ASR | ASR R1, R2, #5 | R[1] = R[2] >> 5 (sign extend) |
| LSR | LSR R1, R2, #5 | R[1] = R[2] >> 5 (zero extend) |
| LSL | LSL R1, R2, #5 | R[1] = R[2] << 5 |

**Control Flow Instructions**

|  |  |  |
| --- | --- | --- |
| CMP | CMP R1, R2 | Set Cond. Flags using R[1] - R[2] |
| B | B Label | Unconditionally goto Label |
| BL | BL Label | Unconditionally goto Label  R[14] = PC+4 |
| BX | BX R14 | PC = R[14] |
| BEQ | BEQ Label | Branch to Label if Equal |
| BNE | BNE Label | Branch to Label if Not Equal |
| BLE | BLE Label | Branch to Label if Less Than or Equal |
| BLT | BLT Label | Branch to Label if Less Than |
| BGE | BGE Label | Branch to Label if Greater Than or Equal |
| BGT | BGT Label | Branch to Label if Greater Than |

**Data Transfer Instructions**

|  |  |  |
| --- | --- | --- |
| **Base** | | |
| LDR/LDRH/LDRB | LDR R1, [R0] | R[1] = Mem[ R[0] ]; |
| LDRSH/LDRSB | LDRSH R1, [R0] | R[1] = Mem[ R[0] ]; sign extend value |
| STR/STRH/STRB | STR R1, [R0] | Mem[ R[0] ] = R[1] |
| PUSH | PUSH {R4-R6, LR, FP} | Store the contents of registers R4 through R6, LR and FP on the stack. Update the stack pointer. |
| POP | POP {R4-R6, LR, FP} | Load from the stack into registers R4 through R6, LR and FP. Update the stack pointer. |
| **Base + Displacement** | | |
| LDR/LDRH/LDRB | LDR R1, [R0, #4] | R[1] = Mem[ R[0] + 4 ]; |
| LDRSH/LDRSB | LDR R1, [R0, #4] | R[1] = Mem[ R[0] + 4 ]; sign extend value |
| STR/STRH/STRB | STR R1, [R0, #4] | Mem[ R[0] + 4 ] = R[1] |

**Registers**

|  |  |  |
| --- | --- | --- |
| Name | Use | Preserved |
| R0 | General Purpose, Argument, and Return Value | No |
| R1-R3 | General Purpose | No |
| R4-R10 | General Purpose | Yes |
| R11 (FP) | Frame Pointer | Yes |
| R12 | Intra Proc. Call | No |
| R13(SP) | Stack Pointer | Yes |
| R14(LR) | Link Register | Yes |
| R15(PC) | Program Counter | No |
| CPSR | Flags | No |

|  |  |  |
| --- | --- | --- |
| **revision** |  |  |
| **v1** | **changed sdr to str** |  |
| **v2** | **list link register as preserved** |  |