* A, B i D corresponen als continguts dels registres del GPBR rs, rt i rd respectivament

|  |  |  |  |
| --- | --- | --- | --- |
| **Data Transfer Instructions** | | | |
| Load Byte | RI | LDB rs rd imm | DI ⟵ (M[imm + AI]7)24 ##M[imm + AI]7..0 |
| Load Byte Unsigned | RI | LDBU rs rd imm | DI ⟵ (0)24 ##M[imm + AI]7..0 |
| Load Halfword | RI | LDH rs rd imm | DI ⟵ (M[imm+AI]15)16##M[imm + AI]15..0 |
| Load Halfword Unisigned | RI | LDHU rs rd imm | DI ⟵ (0)16 ##M[imm + AI]7..0 |
| **Load Word** | RI | LD rs rd imm | DI ⟵ M[imm + AI]31..0 |
| Load Float | RI | LDF rs rd imm | DFP ⟵ M[imm + AI]31..0 |
| Store Byte | RI | STB rs rd imm | M[imm + AI]7..0 ⟵ DI 7..0 |
| Store Halfword | RI | STH rs rd imm | M[imm + AI]15..0 ⟵ DI 15..0 |
| **Store Word** | RI | ST rs rd imm | M[imm + AI]31..0 ⟵ DI |
| Store Float | RI | STF rs rd imm | M[imm + AI]31..0 ⟵ DFP |
| Move Integer 2 FP | RI | MI2F rs rd | DFP ⟵ (float)AI |
| Move FP 2 Integer | RI | MF2I rs rd | DI ⟵ (int)AFP |

|  |  |  |  |
| --- | --- | --- | --- |
| **Arithmetic, Logical Instructions** | | | |
| **Addition** | R3 | ADD rd rs rt | DI ⟵ AI + BI |
| **Add** **immediate** | RI | ADDI rd rs imm | DI ⟵ AI + imm |
| **Substraction** | R3 | SUB rd rs rt | DI ⟵ AI – BI |
| Multiplication | R3 | MULT rd rs rt | DI ⟵ AI \* BI |
| Division | R3 | DIV rd rs rt | DI ⟵ AI / BI |
| **And** | R3 | AND rd rs rt | DI ⟵ AI & BI |
| **Or** | R3 | OR rd rs rt | DI ⟵ AI | BI |
| **XOR** | R3 | XOR rd rs rt | DI ⟵ AI ⊕ BI |
| Negate | R3 | NEG rd rs | DI ⟵ !AI + 1 |
| **Not** | R3 | NOT rd rs | DI ⟵ !AI |
| Integer to 7seg | R3 | I2SS rd rs | DI ⟵ (4 dig. 7sg conv)AI |
| Load High part register | RI | LHI rd imm | DI ⟵ imm ## DI 15..0 |
| Load Low part register | RI | LLO rd imm | DI ⟵ DI 31..16 ## imm |
| **Shift Left** | R3 | SL rd rs rt | DI ⟵ (AI << BI) ## (0)B |
| **Shift Right (Logical)** | R3 | SR rd rs rt | DI ⟵ (0)B ## (AI >> BI) |
| Shift Right (Arithmetic) | R3 | SRA rd rs rt | DI ⟵ (AI 31)B ## (AI >> BI) |
| Compare (equal) | R3 | EQU rd rs rt | DI ⟵ (AI == BI) ? (1)32 : (0)32 |
| **Compare (greater)** | R3 | CMP rd rs rt | DI ⟵ (AI > BI) ? (1)32 : (0)32 |
| Compare (lower) | R3 | LOT rd rs rt | DI ⟵ (AI < BI) ? (1)32 : (0)32 |

|  |  |  |  |
| --- | --- | --- | --- |
| **Floating Point Arithmetic Instructions** | | | |
| Float Addition | R3 | FADD rd rs rt | DFP ⟵ AFP + BFP |
| Float Substraction | R3 | FSUB rd rs rt | DFP ⟵ AFP – BFP |
| Float Multiplication | R3 | FMULT rd rs rt | DFP ⟵ AFP \* BFP |
| Floating Point Division | R3 | FDIV rd rs rt | DFP ⟵ AFP / BFP |
| Float “greater than” comp. | R3 | FGRT rd rs rt | DI ⟵ (AFP > BFP) ? (1)32 : (0)32 |

|  |  |  |  |
| --- | --- | --- | --- |
| **Control Instructions** | | | |
| **Branch if equal** | RI | BEQ rs rd imm | if (AI == DI) PC ⟵ PC + imm |
| Branch if float equal | RI | BFEQ rs rd imm | if (AFP == DFP) PC ⟵ PC + imm |
| **Jump** | RJ | JMP imm | PC ⟵ PC + imm |
| Call | RJ | CALL imm | PC ⟵ PC + imm  LIFO.ADD(PC) |
| Return from call | RJ | RETURN | PC ⟵ LIFO.POP() |

\*Important: per a fer el retorn cal afegir la LIFO i la opció de retornar en R21 el valor del PC anterior.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Data Transfer Instructions** | | | | |
| **Instruction** | **Type** | **Opcode** | **Flags** | **Flags 2** |
| Load Byte | RI | 0111 | 10 | - |
| Load Byte Unsigned | RI | 0110 | 10 | - |
| Load Halfword | RI | 0111 | 01 | - |
| Load Halfword Unisigned | RI | 0110 | 01 | - |
| **Load Word** | RI | 0111 | 00 | - |
| Load Float | RI | 0111 | 11 | - |
| Store Byte | RI | 1000 | 10 | - |
| Store Halfword | RI | 1000 | 01 | - |
| **Store Word** | RI | 1000 | 00 | - |
| Store Float | RI | 1000 | 11 | - |
| Move Integer 2 FP | RI | 0100 | 01 | - |
| Move FP 2 Integer | RI | 0100 | 00 | - |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Arithmetic, Logical Instructions** | | | | |
| **Instruction** | **Type** | **Opcode** | **Flags** | **Flags 2** |
| **Addition** | R3 | 0000 | 00 | 00000000000 |
| **Add** **immediate** | RI | 0000 | 01 | - |
| **Substraction** | R3 | 0000 | 00 | 00000000001 |
| Multiplication | R3 | 0000 | 00 | 00000000010 |
| Division | R3 | 0000 | 00 | 00000000011 |
| **And** | R3 | 0001 | 00 | 00000000000 |
| **Or** | R3 | 0001 | 00 | 00000000001 |
| **XOR** | R3 | 0001 | 00 | 00000000010 |
| Negate | R3 | 0001 | 00 | 00000000011 |
| **Not** | R3 | 0001 | 00 | 00000000100 |
| Integer to 7seg | R3 | 1111 | 00 | 00000000000 |
| Load High part register | RI | 0010 | 00 | - |
| Load Low part register | RI | 0010 | 01 | - |
| **Shift Left** | R3 | 0011 | 00 | 00000000000 |
| **Shift Right (Logical)** | R3 | 0011 | 01 | 00000000000 |
| Shift Right (Arithmetic) | R3 | 0011 | 01 | 00000000001 |
| Compare (equal) | R3 | 0101 | 00 | 00000000000 |
| **Compare (greater)** | R3 | 0101 | 00 | 00000000001 |
| Compare (lower) | R3 | 0101 | 00 | 00000000010 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Floating Point Arithmetic Instructions** | | | | |
| **Instruction** | **Type** | **Opcode** | **Flags** | **Flags 2** |
| Float Addition | R3 | 0000 | 10 | 00000000000 |
| Float Substraction | R3 | 0000 | 10 | 00000000001 |
| Float Multiplication | R3 | 0000 | 10 | 00000000010 |
| Floating Point Division | R3 | 0000 | 10 | 00000000011 |
| Float “greater than” comp. | R3 | 0101 | 10 | 00000000001 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Control Instructions** | | | | |
| **Instruction** | **Type** | **Opcode** | **Flags** | **Flags 2** |
| **Branch if equal** | RI | 1101 | 00 | - |
| Branch if float equal | RI | 1101 | 10 | - |
| **Jump** | RJ | 1100 | 00 | - |
| Call | RJ | 1100 | 01 | - |
| Return from call | RJ | 1100 | 10 | - |

# Mides i constants del sistema

Format <origen><identificador>

El format consisteix en dues úniques lletres majúscules, on l’origen fa referència al bloc general (o tipus) de mides a la que pertany la dada, mentre que l’identificador s’encarrega de distingir els diferents valors dins de cada bloc.

* Bus **B**
  + Adreça **BA**
  + Dada **BD**
  + Error **BE**
* Sistema **S**
  + Adreça de sistema **SA**
  + Adreça de GPBR **SB**
  + Dada **SD**
  + Instruction **SI**
  + Info ins. compressed **SX**
* Instrucció **I**
  + Adreça de GPBR **IB**
  + Opcode **IO**
  + Flag 1 **IX**
  + Flag 2 **IY**
  + Jump extended **IJ**
* Floating Point **F**
  + Dada **FD**
  + Mantissa **FM**
  + Exponent **FE**
* Instruction Cache Memory **C**
  + Base @ **CA**
  + Size **CS**
* LIFO **L**
  + Size **LS**