КОМАНДЫ ПРОЦЕССОРА

В процессоре присутствуют:

* 12 архитектурных 32-ух разрядных регистров общего назначения:
  + A, B, C, D, E, F, G, H – регистры для хранения состояний хеш-суммы;
  + W – регистр для хранения слова данных для одной итерации;
  + K – регистр для хранения константы;
  + MR (Memory Register) – регистр для работы с памятью;
  + LR (Link Register) – регистр для хранения адреса возврата из подпрограммы;
  + zero – регистр, аппаратно хранящий число ноль;
  + t0, t1, t2 – регистры для хранения временных переменных.
* регистр программного счётчика;
* регистр конфигурации процессора (не будет задействован в текущей реализации).

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Таблица №1. Типы инструкций. | | | | | |
| Тип | Формат | | | | |
| R | opcode (4 бита) | rs (4 бита) | | rt (4 бита) | rd (4 бита) |
| M | opcode (4 бита) | mode (2 бита) | address (10 бит) | | |

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Таблица №2. Кодировка инструкций. | | | | | | | | | | | | |
| описание | | | ассемблерный код | Формат инструкции (всего 16 бит) | | | | | | | | |
| кодировка  (4 бит) | | | | некоторые аргументы инструкции (12 бит) | | | | |
| сложение | R | R[rd]R[rs]+R[rt] | ADD rd rs rt | 0 | 0 | 0 | 0 | rs | | | rt | rd |
| сложение с константой | R | R[rd]R[rt]+ExtImm | ADDI rd rt imm | 0 | 0 | 0 | 1 | imm | | | rt | rd |
| Побитовое ИЛИ | R | R[rd]R[rs]|R[rt] | OR rd rs rt | 0 | 0 | 1 | 0 | rs | | | rt | rd |
| Побитовое И | R | R[rd]R[rs]&R[rt] | AND rd rs rt | 0 | 0 | 1 | 1 | rs | | | rt | rd |
| Побитовое исключающее ИЛИ | R | R[rd]R[rs]^R[rt] | XOR rd rs rt | 0 | 1 | 0 | 0 | rs | | | rt | rd |
| Побитовое отрицание | R | R[rd]~(R[rs]|R[rt]) | NOR rd rs rt | 0 | 1 | 0 | 1 | rs | | | rt | rd |
| Логический сдвиг влево | R | R[rd]R[rs]<<R[rt] | SLL rd rs rt | 0 | 1 | 1 | 0 | rs | | | rt | rd |
| Циклический сдвиг влево | R | R[rd]R[rs]<<<R[rt] | ROT rd rs rt | 0 | 1 | 1 | 1 | rs | | | rt | rd |
| Условное ветвление, если не равны | R | if(R[rs]!=R[rt]) PCallign{PC+R[rd]} | BNE rd rs rt | 1 | 0 | 0 | 0 | rs | | | rt | rd |
| загрузка из памяти в регистр | M | MR[15:0MEM[addr] | LDL addr | 1 | 0 | 0 | 1 | 0 | \* | addr | | |
| M | MR[31:16]MEM[addr] | LDH addr | 1 | 0 | 0 | 1 | 1 | \* | addr | | |
| сохранение из регистра в память | M | MEM[addr]MR[15:0] | STL addr | 1 | 0 | 1 | 0 | 0 | \* | addr | | |
| M | MEM[addr]MR[31:16] | STH addr | 1 | 0 | 1 | 0 | 1 | \* | addr | | |
| безусловный переход по адресу | M | PCJumpAddr | JMP addr | 1 | 0 | 1 | 1 | 0 | 0 | addr | | |
| вызов подпрограммы по адресу | M | LR PC + 2; PCJumpAddr | JAL addr | 1 | 0 | 1 | 1 | 0 | 1 | addr | | |
| безусловны переход по значению регистра | M | PCR[rs] | JR rs | 1 | 0 | 1 | 1 | 1 | 0 | rs | | |
| вызов подпрограммы по значению регистра | M | LRPC+2;PCR[rs] | JALR rs | 1 | 0 | 1 | 1 | 1 | 1 | rs | | |
| Нет операции |  | PCPC | NOP | 1 | 1 | 0 | 0 |  | | | | |
|  |  |  |  | 1 | 1 | 0 | 1 |  |  |  | | |
|  |  |  |  | 1 | 1 | 1 | 0 |  |  |  | | |
|  |  |  |  | 1 | 1 | 1 | 1 |  |  |  | | |

Темные ячейки – безразличные, т.е. их значение может быть любым и никак не влияет на результаты команд

* – расширение адреса для выбора адресного пространства (формально третий тип инструкции)

ExtImm={28’b0,instr[11:8]}

Align{} =

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Таблица 3. Кодировка регистров | | | | |
| регистр (название в ассемблере) | номер регистра | | | |
| A | 0 | 0 | 0 | 0 |
| B | 0 | 0 | 0 | 1 |
| C | 0 | 0 | 1 | 0 |
| D | 0 | 0 | 1 | 1 |
| E | 0 | 1 | 0 | 0 |
| F | 0 | 1 | 0 | 1 |
| G | 0 | 1 | 1 | 0 |
| H | 0 | 1 | 1 | 1 |
| W | 1 | 0 | 0 | 0 |
| K | 1 | 0 | 0 | 1 |
| MR | 1 | 0 | 1 | 0 |
| LR | 1 | 0 | 1 | 1 |
| zero | 1 | 1 | 0 | 0 |
| t0 | 1 | 1 | 0 | 1 |
| t1 | 1 | 1 | 1 | 0 |
| t2 | 1 | 1 | 1 | 1 |
|  | | | | |