**ШИНЫ**

Адресная шина (АВ)

|  |  |
| --- | --- |
| 2 | 12 |
| Номер устройства | Адрес |

Первые два бита отвечают за выбор устройства

0 – ничего

1 – RAM

2 – RCM

3 - СТЕК

Контрольная шина (СВ)

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  | 1 | 1 | 1 |
|  |  |  |  |  |  |  | 0 – ничего  1 – записать в AR с шины данных | 0 – ничего  1 – выставить AR на шину адреса | 0 – чтение  1 - запись |

**RAM**

Чтение происходит за два такта. Первый такт сменяется адрес, второй так – на шину выставляются правильно прочитанные данные.

**СТЕК**

Стек активируется при выставлении старших бит адресной шины (AB[13..12]) в 1.

Микрокомандами для стека служат младшие биты адресной шины:

|  |  |
| --- | --- |
| Младшие биты | Микрокоманда |
| 0 | - |
| 1 | Увеличить указатель стека |
| 2 | Уменьшить указатель стека |
| 3 | Записать данные в стек |
| 4 | Прочитать данные из стека |
| 5 | Прочитать регистр флагов из стека |

**Внутренняя управляющая шина (ICB) и внутренние шины блоков.**

Контролирует направление связи всех внутренних шин с основными. Внутренняя память блока, содержащего микрокоманды имеет ширину 48 бит: ICB[15..0] CB[15..0] AB[15..0].

Младший байт ICB управляет связями всех внутренних шин. ICB[7..0]

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Память микрокоманд в шины блока | | Шины блока в общие шины | | | Общие шины в шины блока | | |
| CB | AB | CB | AB | DB | CB | AB | DB |
| 0 – линия разорвана  1 – линия связана в указанном направлении | | | | | | | |

Старший байт ICB отвечает за различные взаимодействия внутри конкретного блока.

**Извлечение инструкций, Instruction Fetch (IF)**

Блок имеет следующую последовательность операций:

|  |  |  |
| --- | --- | --- |
| Номер ICB | Фронт Срез | Значение |
| 8 | / | Сохранить значение «было ли предыдущее предсказание верным» |
| 9 | / | Обновить jump регистры если предсказание было верным |
| \ | Остановить следующие шаги конвейера, если предсказание было не верным.  Прочитать инструкцию с не предсказанного адреса |
| 10 | / | (прочитается новая инструкция) |
| \ | Обновить все регистры адресов и прыжков |
| 11 | / | Выставить прочитанную инструкцию на выходные шины |

Необходимо учесть, что память с инструкциями (IROM) имеет динамическое управление. Переключение инструкции происходит по фронту каждого такта.

**Контрольный модуль (СМ)**

Включает в себя бесконечный итератор, который проходит следующие шаги:

|  |  |
| --- | --- |
| Номер шага | Значение |
| 0 | Запустить все стадии конвейера |
| 1 | Сохранить результаты каждой стадии |
| 2 | Сменить промежуточные регистры |

**Промежуточные регистры**

IF -> OF AND EXEC

|  |  |
| --- | --- |
| 63..40 | 39..0 |
| - | Прочитанная инструкция |

OF AND EXEC -> WRITE

|  |  |  |
| --- | --- | --- |
| 55..48 | 47..32 | 31..0 |
| Флаги операции | Результат операции | Операнды команды |

**Архитектура системы команд (АСК)**

Команда вместе с операндами занимает 40 бит

|  |  |  |
| --- | --- | --- |
| 39..32 | 31..16 | 15..0 |
| Команда | Операнд 1 | Операнд 2 |

Биты команды имеют следующий формат

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 7 | 6 | 5 | 4 | 3..0 |
| Команда является прыжком | 2 операнда | 1 операнд | 0 операндов | Номер команды |

Полный список реализуемых команд

|  |  |  |  |
| --- | --- | --- | --- |
| Обозначение | Битовое представление | HEX представление | DEC представление |
| NOTZ | 00100001 | 21 | 33 |
| AND | 01000010 | 42 | 66 |
| INCS | 00100011 | 23 | 35 |
| ROL | 00100100 | 24 | 36 |
| MOV | 00100101 | 25 | 37 |
| JMP | 10010110 | 96 | 150 |
| PUSH | 00100111 | 27 | 39 |
| POP | 00011000 | 18 | 24 |
| HLT | 00011001 | 19 | 25 |
| JBZ | 10011010 | 9A | 154 |

**Извлечение операндов (OF)**

В блоке извлечения операндов находится два регистра, отвечающие за временное сохранение операндов.

Полная программа блока включает в себя загрузку обоих операндов. Управление количеством загружаемых операндов происходит за счет изменения длины выполняемой микропрограммы (константа длины для внутреннего итератора).

Даже если длина программы равна 0, то на выходе ready имеется мгновенный прыжок (101), который позволяет использовать этот выход для запуска следующего блока выполнения.

Внутренняя шина блока (ICB[15..8])

|  |  |
| --- | --- |
| Номер бита | Значение |
| 8 | Выставить на шину адреса адрес OP1 |
| 9 | Выставить на шину адреса адрес OP2 |
| 10 | Сохранить OP1 |
| 11 | Сохранить OP2 |

Микропрограмма извлечения операндов

|  |  |  |
| --- | --- | --- |
| ICB | CB | AB |
| Ничего не делаем | | |
| 00000000\_11000000 | 00000000\_00000000 | 00000000\_00000000 |
| Выставляем адрес первого операнда на шину | | |
| 00000001\_10110001 | 00000000\_00000000 | 00000000\_00000000 |
| Читаем первый операнд и сохраняем его | | |
| 00000101\_10110001 | 00000000\_00000000 | 00000000\_00000000 |
| Читаем второй операнд | | |
| 00000010\_10110001 | 00000000\_00000000 | 00000000\_00000000 |
| Читаем второй операнд и сохраняем его | | |
| 00001010\_10110001 | 00000000\_00000000 | 00000000\_00000000 |

**Регистр флагов (FR)**

Имеет адрес 0 в пространстве регистров.

|  |  |
| --- | --- |
| Бит | Значение |
| 0 | 0 – (res != 0) 1 – (res == 0) |
| 1 | 0 – (res > 0) 1 – (res < 0) |
| 2 | Carry бит |
| 3 | Бит переполнения |
| 4 | Стек пуст ошибка чтения |
| 5 | Стек полон ошибка записи |
| 6 |  |

**Операции**

**AND (DEC 66)**

|  |  |  |
| --- | --- | --- |
| ICB | CB | AB |
| Ничего не делаем | | |
| 00000000\_11000000 | 00000000\_00000000 | 00000000\_00000000 |
| Защелкиваем входные данные | | |
| 80000000\_11000000 | 00000000\_00000001 | 00100000\_00000000 |