НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ

«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ»

КАФЕДРА ОБЧИСЛЮВАЛЬНОЇ ТЕХНІКИ

Лабораторна робота №6

з дисципліни **«**Комп’ютерна схемотехніка**»**

Виконав:

студент 3 курсу

ФІОТ гр. ІО-21

Кузьменко Володимир

Київ – 2014 р.

**Задание:** На PLMT с параметром и ранее разработанных ICTR, RAM, FM, LSM и AU разработать вычислитель, реализующий команды: сложения, вычитания, пересылок (RAM ↔ FM), условных и безусловных переходов, команды ввода-вывода. Ранее разработанную программу выполнения многотактной операции модернизировать для случая, когда операнды находятся в RAM и результат должен быть дописан в RAM.

**Выбор структуры схемы CPU.**

Этот выбор во многом определяется параметрами и структурой RAM, FM и AU. Структурная схема показана как на рисунке 1.



Рисунок 1. – Структурная схема CPU.

Пояснения к рисунку:

RAM – запоминающее устройство;

WRR – сигнал записи в RAM;

FM – быстрая память;

AP – адрес куда необходимо записать данные из шины P;

AQ – адрес куда необходимо записать данные из шины Q;

WRF– сигнал записи в быструю память;

RRF– сигнал чтения с быстрой памяти;

LSM – схема логического сумматора;

FL3-FL0 – управляющие входы сумматора;

CI – входной перенос сумматора;

CO – выходной перенос сумматора;

SI – знак результата;

RZ – признак нуля результата;

SHU> - схема сдвига на один разряд влево;

Q – значение которое будет записываться в сдвинутый разряд;

P – значение выдвигаемого разряда;

F – управляющий сдвигателем сигнал;

MUX – мультиплексор;

DI – данные для записи в FM или RAM;

C1-C0 – управляющие входы мультиплексоров;

ICTR – счетчик команд;

FIC3-FIC0 – управляющие сигналы счетчика;

WRIC – запись в счетчик;

RIC – сброс счетчика в ноль;

IRG – регистр команд;

COP – блок управления операциями;

CCRG – регистр признаков операции.

**Выбор и кодирование системы команд CPU.**

По заданию у нас 8-битный CPU, в котором содержится RAM емкостью 213 слов, FM емкостью 11 слов и LSM на 10 микроопераций.

Согласно заданию, CPU должен реализовывать арифметически (АО) и логические (LO) операции, команды пересылки слов из FM в RAM (FR), и обратно (RF), условные (JC) и безусловные (JMP) переходы и команды ввода (IN) и вывода (OUT). В командах AO и LO надо задать 5 бита кода МО LSM (4 бита LSM и 1 бит SHU>) и 2 4-битных адреса операндов в FM, что в сумме составит: 25×24×24=213 разновидностей таких операций.

В командах FR и RF надо указать направление пересылки, адрес ячейки в FM и в RAM, что дает: 2×210×213=224.

Команды JC осуществляют нарушение естественного порядка следования команд при выполнении заданного условия. Если предположить, что таких условий (RZ – признак нуля, SI – знак, CO – перенос в следующий разряд, RO – выдвигаемый разряд) четыре, а переход может осуществляться по выполнению или невыполнению условий, то разновидностей команд JC будет: 8×213=216 разновидностей.

Команда JMP имеет 213 разновидностей.

Команды ввода-вывода IN и OUT задают направление передачи данных (в CPU или из CPU) и номер внешнего устройства. Если предположить, что количество таких устройств будет не более чем емкость RAM, то получим: 2×213=214 разновидностей таких команд.

Для задания кода команды необходимо 3 байта, то есть 24 бита. Закодированные команды представлены в Таблице 1.

Таблица 1. – Кодированные команды CPU.

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Операция | | Byte2 | | | Byte1 | | | Byte0 | |
| 7 6 5 4 3 2 1 0 | | | 7 6 5 4 3 2 1 0 | | | 7 6 5 4 3 2 1 0 | |
| AO | 0  1  CO  SI  P | 0 0 0 0 0 0  0 0 0 0 0 1  0 0 0 0 1 0  0 0 0 0 1 1  0 0 0 1 0 0 | | F3 F2 F1 F0  F3 F2 F1 F0  F3 F2 F1 F0  F3 F2 F1 F0  F3 F2 F1 F0 | | AI | | AO1 | AO2 |
| LO | | 0 0 1 0 0 0 | | F3 F2 F1 F0 | | AI | | AO1 | AO2 |
| FR  RF | | 0 1 0 0 0 0 1 0 1  0 1 0 1 1 1 1 0 1 | | | | A FM | | A RAM | |
| JC | | 1 0 1 | A CC | A RAM | | |  | | |
| IN  OUT | | 1 1 0 0 0 0  1 1 0 1 1 1 | | A RAM | | |  | | |
| JMP | | 1 0 0 0 0 0 | | A RAM | | |  | | |
| SHU> | 0  1  CO  SI  P | 1 1 1 0 0 0 1 0 1 0 0  1 1 1 0 0 1 1 0 1 0 0  1 1 1 0 1 0 1 0 1 0 0  1 1 1 0 1 1 1 0 1 0 0  1 1 1 1 0 0 1 0 1 0 0 | | | | A FM |  | | |

Во втором байте три старших разряда используются как кодирование семи главных операций: AO, LO, FR и RF, IN и OUT, JMP, JC и SHU>.

В командах AO и LO необходимо код операции в LSM, адреса операндов и адрес ячейки FM, в которую необходимо записать.

При выполнении AO необходимо управлять переносом в младший разряд. При сдвигах влево, которые выполняются путем сложения D + D = 2D, необходимо управлять вдвигаемым разрядом. Отсюда следует, что, в общем случае, при выполнении AO в младший разряд LSM могут быть из AO может иметь 5 разновидностей (0, 1, CO, SI, QO), в зависимости от того, что подается в младший разряд LSM по цепи переноса CI.

LO не требует указания кода символа, поступающего на CI, так как переносы при выполнении LO блокируются.

Операции сдвига вправо (SHU>) являются одноадресными, так как операнд Q извлекается из FM, сдвигается и в него можно записать 0, 1, CO, SI, QO. Отсюда следует, что разновидностей операций будет 5. Если для кодирования вдвигаемого бита использовать те же комбинации цифр, что и при выполнении AO, то всю необходимую информацию для сдвига можно упаковать в два байта.

Будем предполагать, что ввод и вывод осуществляется через фиксированный регистр RG в FM (например, 0-й).

**Проектирование ICTR.**

Так как у нас используется 24-х разрядная команда, то необходимо построить 24-х разрядный регистр команд. А так как у нас в памяти хранятся слова по 8 бит, то этот регистр должен записывать в три этапа: нулевой байт, первый байт и второй байт.



Рисунок 2. – Схема одного разряда IRG.

**Проектирование регистра признаков операции.**

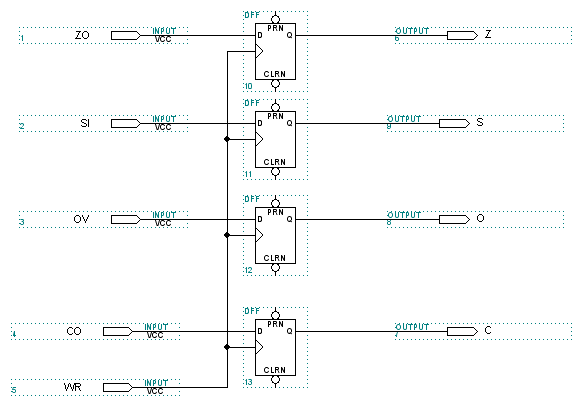


Рисунок 3. – Общая схема регистра признаков.

**Алгоритм деления по первому способу.**

так

ні

1

0

Початок

Rz := 0..01М;

Rx := X2;

Ry := Y2;

Rx[1]

Rx := Rx + Ry;

Rx := l(Rx).0;

Rz := l(Rx).[1];

Rz[1]

Кінець

Рисунок 4 – Алгоритм деления

Rx := Rx + + c;

**Микропрограмма деления по первому способу.**

Z=X/Y.

Пусть Y находится в RAM[0], X в RAM[1], Z в FM[2]. А команда лежит с RAM[11]

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Byte2 | Byte1 | Byte0 |  |
|  | 7 6 5 4 3 2 1 0 | 7 6 5 4 3 2 1 0 | 7 6 5 4 3 2 1 0 |  |
| 11 | 0 1 0 1 1 1 1 0 | 1 0 0 0 0 0 0 0 | 0 0 0 0 0 0 0 0 | Пересылка Y из RAM в FM |
| 12 | 0 1 0 1 1 1 1 0 | 1 0 0 0 0 1 0 0 | 0 0 0 0 0 0 0 1 | Пересылка X из RAM в FM |
| 13 | 1 0 1 1 0 0 0 0 | 0 0 0 1 0 0 0 1 |  | Если RX[1] = 0, то на (15) |
| 14 | 0 0 0 0 0 0 0 1 | 1 0 0 0 1 0 0 0 | 0 1 0 0 0 0 0 0 | Сумма X + Y |
| 15 | 0 0 0 0 0 0 0 1 | 1 0 0 0 1 0 0 0 | 0 1 0 0 0 0 0 0 | Сумма X + Y + C |
| 16 | 0 0 0 0 0 0 0 1 | 1 0 0 0 1 0 0 0 | 0 1 0 0 0 0 1 0 | Сдвиг влево X |
| 17 | 0 0 0 0 1 0 0 1 | 1 0 0 0 0 1 0 0 | 0 0 1 0 0 0 0 1 | Сдвиг влево Z |
| 18 | 1 0 1 0 0 0 0 0 | 0 0 0 0 1 1 0 1 |  | Если RZ[1] = 1, то на (13) |
| 19 | 0 1 0 0 0 0 1 0 | 1 0 0 0 0 1 0 0 | 0 0 0 0 0 0 0 1 | Запись в RAM старших разрядов |
| 20 | 0 1 0 0 0 0 1 0 | 1 0 0 0 1 0 0 0 | 0 0 0 0 0 0 1 0 | Запись в RAM младших разрядов |

**Цифровая диаграмма.**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| № | Rz | Rx | Ry | Коментар |
| П.С. | 00000001 | 10100100 | 10100001 | Y2 ДК = 01011111 |
| 1 | 00000010 | 10100100  +001011111  00000011  00000110 |  | +  ⮘ |
| 2 | 00000101 | 00000110  +110100001  10100111  01001110 |  | +  ⮘ |
| 3 | 00001011 | 01001110  +001011111  10101101  01011010 |  | +  ⮘ |
| 4 | 00010111 | 01011010  +001011111  10111001  01110010 |  | +  ⮘ |
| 5 | 00101110 | 01110010  +001011111  11010001  10100010 |  | +  ⮘ |
| 6 | 01011100 | 10100010  +10100001  01000011  10000110 |  | +  ⮘ |
| 7 | 10111001 | 10000110  +110100001  00100111  01001110 |  | +  ⮘ |
| 8 | 01110011 | 01001110  +01011111  10101101  01011010 |  | +  ⮘ |