Задание

Найти наименьший отчёт, зарегистрированный в «электронном журнале» во время предпоследнего сеанса снятия показаний датчиков.

Исходный данные:

Количество датчиков N – 5

Количество сеансов n – 2

Начальный адрес для файла G Аж.нач. – 2225(10) = 08B1(16)

Начальный адрес программы Апр.нач. – FA25(16)

Микропроцессорная система обработки журнальных данных

1. Каждому включению МСОЖД должен предшествовать рабочий цикл некоторого автомата для сбора данных (АСД), который в течении своего суточного цикла (когда сигнал X1 = 1) организует n сеансов связи и во время каждого «сеанса» поочерёдно (в порядке возрастания номеров) подключает датчики Д1…Дn ко входу данных Д ОЗУ, формирует адрес (номер) Aij ячейкидля записи показаний Дij i-го датчика в j-ом «сеансе» связи и вырабатывает саму команду w запись. В результате в конце суточного цикла сбора данных, когда сигнал 1|0 сменит свой уровень с 1 на 0, в ОЗУ окажется сформированным готовым к обработке файл G данных (т.е. электронный журнал), структура которого полностью известна.

На рисунке 1 изображена упрощённая структура такого автомата для сбора данных (АСД): в его состав входят программируемый электронный таймер, 8-разрядный N-канальный мультиплексор и два детектора, которые, анализируя показания таймера, формируют сигналы «сеанс» (=1 во время сеанса) и «цикл» (=1 во время цикла). Последний из них X1 хранится в регистре параллельного действия RG, т. е. в «порте» устройства ввода УВ №1 МСОЖД.

Мультиплексор каналов

+

счётчик – формирователь адреса

Детектор «нач/конец сеанса»

Детектор «нач/конец сеанса»

D ОЗУ

(файл G)

A

W/R

RG

D0 (порт УВв1)

Д1

Д1

ДN

(8)

(8)

Канал 1

Канал 2

Канал N-1

(8)

(16)

X2 («сеанс»)

X1 («цикл»)

(8)

W

gt

Рис. 1. Автомат для сбора данных.

2. Описание МСОЖД. Один из возможных вариантов построения МСОЖД показан на рисунке 2, где приведена микропроцессорная система с трёхшинной организацией, а в качестве центрального процессора (CPU) использована БИС КР580ВМ80. Если в ОЗУ в область свободную от обрабатываемого файла G, разместить программу, реализующую заданный пользователем алгоритм обработки, то становится возможной автоматизация процедуры обработки. Программ запускается по окончании суточного цикла сбора данных, а обработка начинается по сигналу X = 0, считываемому с «детектора цикла» АСД и вводимому через устройство ввода УВв №1 по команде INPUT программы. В ходе обработки программа должна обеспечивать неоднократное извлечение из ОЗУ любого элемента файла G, выполнение над ним необходимых арифметических и логических действий, приводящих к формированию конечного результата F, и выдачу F через устройство вывода УВыв №1 пользователю (по команде UOTPUT). Нужно учесть, что структура файла G такова (поочередная запись показаний датчиков в порядке возрастания их номеров во время каждого сеанса), что информация gij, принятая по i-ому каналу (i=1,2,…,n-1) во время j-ого сеанса (j=0,1,2,…,n-1), будет храниться в ячейке ОЗУ с номером = адресом Aij = Aнач + j\*N+i.

А область памяти, которую занимает журнал, находится в диапазоне адресов от Анач до (Анач + n\*N+1).

CPU



A

64К х8

ОЗУ

R/W D R/W

УВв №1

Регистр=порт

УВыв №1

Регитр=порт

ША 16 - разрядная

ШД 8 – разрядная двунаправленная

ШУ 10 - разрядная

Х1

F

MEMW

MEMR

IOR

IOW

Рис. 2. МП – система автоматической ОЖД.

Если в программе выполняется команда, требующая обращение к памяти, то на 16-разрядную шину адреса ША выставляется номер требуемой ячейки Aij, а на 10-разрядной шине управления ШУ формируются сигналы «читай память» MEMR или «пиши в память» MEMW, которые задают режим работы ОЗУ и направление передачи по двунаправленной 8-разрядной шине данных ШД. Если же программа выполнит программу INPUT (или OUTPUT), то на ША выдаётся номер устройства ввода (или вывода), а на ШУ формируются сигналы «ввод/вывод читай» IOR (или «ввод/вывод пиши» IOW).

«Портом» УВв №1 является 8-разрядный регистр-защёлка, у которого младший разряд хранящегося в нем числа определяется сигналом Х1, а сигнал IOR поступает на вход «разрешение выдачи» ОЕ. «Порт» УВыв №1 также является параллельным 8-рязрядным регистром, на тактовый вход которого поступает сигнал IOW. Само Увыв представляет собой 3-разрядный 7-сегментный индикатор (для отображения результата F) и соответствующий кодопреобразователь, включенный между портом и индикатором.

3. Процессор КР580ВМ80.

В состав данного 8-разрядного процессора (рис. 3) входит АЛУ и программно-доступные регистры: аккумулятор РОН А, 6 регистров общего назначения РОН (с «именем» B, C, D, E, H и L) и счётчик команд РС. АЛУ, получив от УУиС указания о типе выполняемой операции f над доставленными на его входе операндами х1 их2, формирует результат F=f(x1 x2) и «флаги» Ф (признаки нулевого – Z, отрицательного – S и чётного результат Р или наличие переноса С7). Существенно, что результат F всегда (по умолчанию) размещается в РОН А. Это означает, что, если содержимое РОН А не является операндом следующей команды, то при программировании её должна предшествовать дополнительная команда (которой нет в алгоритме обработки пользователя) перезаписи содержимого РОН А в свободный РОН или в ячейку М памяти ОЗУ (=Ме-точку). Для программировании операций с РОН каждому из них присвоен порядковый номер i и соответствующий индивидуальный двоичный код ri (табл. 1). Для хранения 16-разрядный чисел d16=adr РОНя могут объединяться в регистровые пары rp: BC=rpB, DE=rpD, HL=rpH.

8-разр. ШД

B……...(8)

C……...(8)D……...(8)E……...(8)H……...(8)

L……...(8)

SP (16)

PC (16)

Pr Agp(16)

РГК

ДШК

УУиС

(УА)

Буф Рг

Буф Рг

Буф. Pr

двунвправ.

f

X2

X1

АЛУ

Рез-т F

z

c

s

p

Блок регистров

Внутренняя 8-разрядная двунаправленная ШД

16 разр. ША

системная

системная

ГТИ

10-разрядная ШУ

системная

**КР 580ВМ80**

Рис. 3. Упрощёшшая модель микропроцессора КР580ВМ80

Нумерация РОН. Табл. 1

|  |  |  |
| --- | --- | --- |
| i | Имя РОН | Код ri |
| 0  1  2  3  4  5  6  7 | B  rp  C  D  rp  E  rp  H  L  M  A | 000  001  010  011  100  101  110  111 |

Указатель стека SP хранит текущий адрес ячейки ОЗУ, являющейся на данный момент вершиной стека.

Счётчик команд PC хранит адрес выполняемой (текущей) команды: после её завершения содержимое PC инкрементируется, т. е.PC(PC) + 1, и через буферный PrАдр выдается на системную ША, а из ОЗУ (по сигналу MEM R) извлекается первый байт <B1> кода следующей команды из ячейки ОЗУ с номером (PC) + 1) и по системной шине ШД передается в регистр команд PrK.

Это означает, что в обычной ситуации процессор может выполнять команды программы только в том порядке, в котором они записаны в ОЗУ.

Чтобы изменить порядок выполнения команд (сделать скачок на несколько ячеек ОЗУ вперёд или назад, организовать ветвление или цикл, выполнить программу), необходимо в программу вставить команду безусловного или условного перехода, которая позволяет скачком менять содержимое (РС) счётчика команд РС.

Все регистры и АЛУ обмениваются между собой 8-разрядными данными d8 через внутреннюю двунаправленную ШД, однако на каждом такте обмен осуществляется только между одной парой «абонентов» (один - отправитель, другой – получатель).

Обмен между внутренней и внешней = системной ШД происходит через двунаправленный буферный регистр.

Каждая (текущая) команда программы (её код находится в PrK, а адрес в РС) выполняется процессором в течении определённого времени, называемого командным циклом продолжительностью от 1 до 4 тактов (при тактовой частоте 2 Mгц). В течении командного цикла устройство управления и синхронизации УуиС, будучи обычным управляющим автоматом (УА), декодирует с помощью дешифратора ДшК первый байт <B1> кода команды и в соответствии с этим кодом на каждом такте вырабатывает сигналы для внутреннего выполнения управлением микроопераций на «избранных» (на данном такте) функциональных узлах (регистрах, АЛУ, селекторе и др.), а также внешние управляющие сигналы (типа MEMR, MEMW, IOW и др.), выдаваемые на системную ШУ.

4. Система команд.

Каждый процессор умеет выполнять ограниченный набор «приказов», входящих в его систему команд. Каждая команда представляет собой многоразрядный двоичный код (от 8 до 24 бит) определённого формата. Для процессора КР580ВМ80 предусмотрены команды трёх форматов: «короткие» однобайтные <B1>, двухбайтные <B1><B2> и трёхбайтные <B1><B2><B3>. Первый байт <B1> команды любого формата содержит код операций Коп, второй <B2> - в двухбайтных командах содержит числовое значение непосредственно задаваемого операнда d8 или порядковый номер n устройства ввода/вывода = port n. Третий и второй байты «длинных» команд содержат либо численное значение 16-разрядного операнда d16, либо 16-разрядный адрес (“adr”): причем старший байт адреса/операнда размещается в <B3>, а младший в <B2>.

Ниже в таблице 2 приведены данные о наиболее ходовых командах процессора.

В таблице 2 поле 1 характеризует формат команды (в байтах), поле 2 – продолжительность командного цикла в количестве тактов; в поле 3 описывается (на языке микрокоманд) выполняемая операция: запись (ri) означает «содержимое» регистра ri, запись <Bi> - «содержимое байта Bi? (HL) – регистровой пары HL, а в поле 5 побитовая структура 1 – го байта <B1> кода команды.

В командах пересылке и загрузки (NN1 и 2) нужно учесть, что при ri = 110 в обмене участвует ячейка М ОЗУ, адрес которой (по умолчанию) хранится в регистровой паре HL. Это означает, что команда MOV и MVI обязательно должна предшествовать команда загрузки регистровой пары (HLadr), т. е. команда LXI.

В ассемблерной записи команд NN3 и 21 фигурирует старший регистр ri (=B, D или H) регистровой пары rp. При выполнении двухоперандных команд (NN7 – 18) первые операнд x1 всегда берётся из РОН А, а второй x2 из другого РОНа или задается непосредственно во втором байте <B2> команды. Логические операции, в отличии от арифметических, выполняются поразрядно.

Для всех команд условного перехода (NN25 – 30) при невыполнении проверяемого условия в PC загружается адрес adr = (PC) + 3.

Система команд процессора КР580ВМ80. Табл.2

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| NN  n/n | Ассемблерная запись команды | Формат  (байты) | | Такты | Выполняемая операция | | Форми-  руемые  флаги | Структура <B1>  (номера разрядов) | | | | | | | |
| Команды пересылке и загрузке | | | | | | | | | | | | | | | |
| 1.  2.  3.  4.  5.  6. | MOV\_ri,rj  MVI\_ri,d8  LXI\_ri,d16  XCHD  PCHL  SPHL | 1  2  3  1  1  1 | 5  7  10  4  5  5 | | | ri← (rj)  ri←B2>  ri←<B2>,ri+1←<B2>  (HL)↔(DE)  PC←(HL)  SP←(HL | нет  нет  нет  нет  нет  нет | 0  0  0  1  1  1 | 1  0  0  1  1  1 | ←  ←  ←  1  1  1 | ri  ri  ri  0  0  0 | →  →  →  1  1  1 | ←  1  0  0  0  0 | ri  1  0  1  0  0 | →  0  1  1  1  1 |
| Арифметические и логические операции | | | | | | | | | | | | | | | |
| 7.  8.  9.  10.  11.  12.  13.  14.  15.  16.  17.  18.  19. | ADD\_ ri  SUB\_ ri  ANA\_ ri  XRA\_ ri  ORA\_ ri  CMP\_ ri  ADI\_ d8  SUI\_ d8  ANI\_ d8  XRI\_ d8  ORI\_ d8  CPI\_ d8  SMA | 1  1  1  1  1  1  2  2  2  2  2  2  1 | 4  4  4  4  4  4  7  7  7  7  7  7  4 | | | A← (A) + (ri)  A← (A) - (ri)  A← (A) /\ (ri)  A← (A) + (ri)  A← (A) \/ (ri)  Сравн. A - (ri)  A← (A) + <B2>  A← (A) - <B2>  A← (A) /\ <B2>  A← (A) + <B2>  A← (A) \/ <B2>  Сравн. A - <B2>  A← (A) | все  все  z, s, p  z, s, p  z, s, p  все  все  все  z, s, p  z, s, p  z, s, p  все  нет | 1  1  1  1  1  1  1  1  1  1  1  1  0 | 0  0  0  0  0  0  1  1  1  1  1  1  0 | 0  0  1  1  1  1  0  0  1  1  1  1  1 | 0  1  0  0  1  1  0  1  0  0  1  1  0 | 0  0  0  1  0  1  0  0  0  1  0  1  1 | ←  ←  ←  ←  ←  ←  1  1  1  1  1  1  1 | ri  ri  ri  ri  ri  ri  1  1  1  1  1  1  1 | →  →  →  →  →  →  0  0  0  0  0  0  1 |
| Операции циклического сдвига и инкремента | | | | | | | | | | | | | | | |
| 20.  21.  22.  23 | INR\_ri  INX\_ri  RLC  RRC | 1  1  1  1 | 5  5  4  4 | | | ri ← (ri) + 1  rp← (rp) + 1  Ai+1←(Ai);A0←(A7);  C7←(A7)  Ai←(Ai+1);A7←(A0);  C7←(A0) | z, s, p  нет  С7  С7 | 0  0  0  0 | 0  0  0  0 | ←  ←  0  0 | ri  ri  0  0 | →  →  0  1 | 1  0  1  1 | 0  1  1  1 | 0  1  1  1 |
| Безусловный переход | | | | | | | | | | | | | | | |
| 24 | JMP\_adr | 3 | 10 | | | PC←<B3><B2> | нет | 1 | 1 | 0 | 0 | 0 | 0 | 1 | 1 |
| Условные переходы | | | | | | | | | | | | | | | |
| 25.  26.  27  28  29  30 | JNZ\_adr  JZ\_adr  JNC\_adr  JC\_adr  JP\_adr  JM\_adr | 3  3  3  3  3  3 | 10  10  10  10  10  10 | | | при z=0  PC←<B3><B2>  при z=1 …..|…..  при С7=0 …..|…..  при С7=1 …..|…..  при S=0 …..|…..  при S=1 …..|….. | нет  нет  нет  нет  нет  нет | 1  1  1  1  1  1 | 1  1  1  1  1  1 | 0  0  0  0  1  1 | 0  0  1  1  1  1 | 0  1  0  1  0  1 | 0  0  0  0  0  0 | 1  1  1  1  1  1 | 0  0  0  0  0  0 |
| Вспомогательные функции | | | | | | | | | | | | | | | |
| 31.  32.  33.  34. | IN\_port n  OUT\_port m  NOP  HLT | 2  2  1  1 | 10  10  4  7 | | | Ввод: А←(port n)  Вывод: А←(port m)  Нет операции  Остановка, стоп | нет  нет  нет  нет | 1  1  0  0 | 1  1  0  1 | 0  0  0  1 | 1  1  0  1 | 1  0  0  0 | 0  0  0  1 | 1  1  0  1 | 1  1  0  0 |

5. Разработка алгоритма

начало

A ← Увв№1

к1

A← (А) ^ maska

к2

Z=0

к3

да

M1:

HL ← Aпред

к4

E ← M[HL]

к5

C ← 0

к6

HL ← (HL)+1

к7

C ← C+1

к8

A ← (C)

к9

Срав.(А) - Кrp

к10

Krp = O5(16) = <B2>

S=0

к11

A ← (E)

к17

УВыв№1 ← (А)

к18

stop

к19

A ← M[HL]

Срав.(А) - E

к13

к12

M2:

S=0

к14

E ← (A)

к15

да (Kt > Krp)

нет (Kt > Krp)

да (gt<gmin)

нет (gt>gmin)

Рис.4. Блок – схема алгоритма обработки журнальных записей.

Алгоритм обработки предписывает поочередное чтение из ОЗУ показаний gt из ячеек с адресами At, попарное сравнение их по величине и запоминание меньшего из них. Блок – схема алгоритма приведена на рис. 4 и содержит 19 команд к1…к19: здесь формируемая величина gmin размещена в РОН Е, текущий адрес Аt = Aпред + ∆At – в регистровой паре HL, а приращение адреса ∆At = kt является параметром цикла обработки и размещается в РОН С; максимальное (=граничное) приращение krp = N = 5 является индикатором окончания цикла анализа данных gt из журнального файла G.

На блок-схеме рис.4:

* команды к1…к3 обеспечивают анализ вводимого с ПЭК сигнала х1 и принятие решения о конце суточного цикла сбора данных и начла обработки файла G; численное значении маски 01(16) гарантирует анализ младшего бита содержимого РОН А;
* команда к4 загружает адрес предпоследнего сеанса Апред в регистровую пару HL;
* команда к5 считывает первый «отсчёт» предпоследнего сеанса g0 из файла G и, записав его в РОН Е, объявляет его равным gmin;
* команда к6 заносит начальное значение Kt = 0 в РОН С; вместе к4…к6 обеспечивают подготовку цикла к7…к15 анализа G;
* команды к7, к8 обеспечивают инкремент текущего адреса At и приращение адреса Kt;
* команды к9 – к11 на основе сравнения kt c krp = 05 обеспечивают выход из цикла обработки и выдачу на индикацию через УВыв№1 рассчитанного значения gmin из РОН Е (к17, к18);
* команда к12 загружает в РОН А текущий отсчёт gt из ячейки M ОЗУ с адресом At (сформированным в регистровой паре HL);
* команды к13 и к14 обеспечивают сравнение текущего отсчёта gt с gmin и при появлении gt<gmin перезапись его в РОН Е (к15);
* команда к16 обеспечивает безусловный переход (БП) на метку М3.

В таблице 3 (в полях 2-6) приведена ассемблерная запись программы, реализующий этот алгоритм. Из неё видно, что для размещения загрузочного модуля (в двоичных кодах) этой программы необходимо 34 байта (ячейки ОЗУ).

Сам загрузочный модуль представлен полями 0 и 1таблицы 3, но для простоты и кратности записан в 16-ричных кодах (h-кодах): коды первых байтов взяты из таблицы 2, а числовые значения <B3> и <B2>, а также начального адреса предпоследнего сеанса FA25(16) для размещения программы в ОЗУ устанавливаются по исходным данным задачи.

Текст программы на ассемблере и её загрузочные модуль (в h-кодах). Табл.3

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Мет-  ки | Адрес ОЗУ | h-коды | NN  коман-  ды | Ассемблер | Комментарий | Бай-  ты | Так-  ты |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 |
| М1 | FA25  FA26 | <B1>DB  <B2>01 | K1 | IN\_00 | A←(port 0) | 2 | 10 |
| FA27  FA28 | <B1>E6  <B2>01 | K2 | ANI\_01 | A←(A)^maska  maska = 01h | 2 | 7 |
| FA29  FA2A  FA2B | <B1>C2  <B2>89  <B3>FE | K3 | JNZ\_M1: | Усл. Переход (УП) по условию Х1 = 0  («обработку начать») | 3 | 10 |
| FA2C  FA2D  FA2E | <B1>21  <B2>86  <B3>26 | K4 | LXI\_H,08B1 | HL←Апред  (Апред = adr = 08B1 | 3 | 10 |
| FA2F | <B1>5E | K5 | MOV\_E,M | E←gt (из ОЗУ) | 1 | 5 |
| FA30  FA31 | <B1>0E  <B2>00 | K6 | MVI\_C,00 | С←0 (At:=0) | 2 | 7 |
| M2 | FA32 | <B1>23 | K7 | INX\_H | HL←(HL)+1  инкремент At | 1 | 5 |
| FA33 | <B1>0C | K8 | INR\_C | C←(C)+1(инкр. Kt) | 1 | 5 |
| FA34 | <B1>79 | K9 | MOV\_A,C | А←С | 1 | 5 |
| FA35  FA36 | <B1>FE  <B2>0D | K10 | CPI\_0C | Сравнение(А) с Krp=05 | 2 | 7 |
| FA37  FA38  FA39 | <B1>F2  <B2>A7  <B3>FE | K11 | JP\_M2: | УП на метку М2  («конец файла G») | 3 | 10 |
| FA3A | <B1>7E | K12 | MOV\_A,M | А←gt (из ОЗУ) | 1 | 5 |
| FA3B | <B1>BB | K13 | CMP\_E | Сравнение (А) и gmin | 1 | 4 |
| FA3C  FA3D  FA3E | <B1>F2  <B2>96  <B3>FE | K14 | JP\_M3: | УП на метку М3  («новый отсчёт gt меньше, чем gmin?») | 3 | 10 |
| FA3F | <B1>5F | K15 | MOV\_E,A | Сохранить gt в РОН E | 1 | 5 |
| FA40  FA41  FA42 | <B1>C3  <B2>98  <B3>FE | K16 | JMP\_M3: | Безусл. Переход (БП)  на метку М3  (PC) ←<B3><B2> | 3 | 10 |
| M2 | FA43 | <B1>7B | K17 | MOV\_A,E | A←(E) | 1 | 5 |
| FA44  FA45 | <B1>D3  <B2>01 | K18 | OUT\_01 | Выдать gmin на индикацию через Увыв №1 | 2 | 10 |
| FA46 | <B1>76 | K19 | HLT | Стоп | 1 | 7 |