

Програмираме паралелен интерфейс

1

Програмираме паралелен интерфейс

Паралелният интерфейс обезпечава едновременно предаване на няколко бита, най-често 8, по съответния брой отделни линии. Използва се когато е необходимо да се обменят голям обем данни с висока скорост.

2

Програмираме паралелен интерфейс

Ако обменът на данни се извършва в зависимост от наличието на синхронизиращ сигнал, тогава се приема, че връзката е синхронна. Ако обменът на данни се извършва в съответствие с появата на сигнали за готовност, потвърждение и други сигнали за състоянието на интерфейса и устройствата за вход-изход, тогава се приема, че обменът е асинхронен.

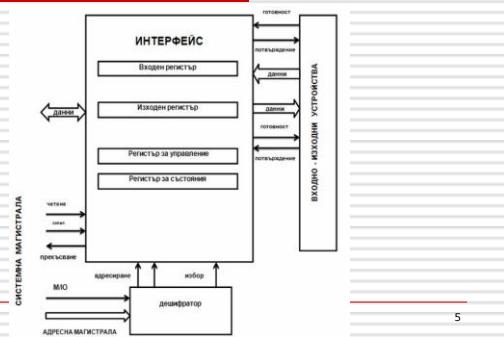
3

Програмираме паралелен интерфейс

Паралелният интерфейс може да бъде разработен с използването на един набор проводници (кабел) за вход и за изход, или по отделни проводници за вход и отделно за изход (като е показано на фиг. 7.1). Като частен случай може да се реализира паралелен интерфейс само за вход или само за изход.

4

Двупосочен паралелен интерфейс



Програмираме паралелен интерфейс

Интерфейсът от фигуранта представлява най-често срещания случай на паралелен интерфейс с отделни проводници за вход и за изход на данни и проводници за готовност и потвърждение.

6

Програмираме паралелен интерфейс

Процедурата при изход на данни е следната:

- В изходно състояние сигнал "готовност" е равен на лог.0, а сигнал "потвърждение" - на лог.1.
- Микропроцесорът, чрез адресната шина и сигнал **M/IO = 0** избира изходния регистър на интерфейса. С управляващ сигнал **WR** данните се записват в този регистър.

7

Програмираме паралелен интерфейс

- Интерфейсът се връща в изходно състояние, при което сигналът "готовност" става равен на лог.0. След него и външното устройство променя сигнала "потвърждение" в лог.1. С това процедурата за изход на един байт данни завършва и интерфейсът е готов за обмен на следващия байт.

9

Програмираме паралелен интерфейс

- Интерфейсът установява сигнал "потвърждение" в лог.1 и генерира заявка за прекъсване, а външното устройство променя сигнал "готовност" в лог.0. Докато сигнал "потвърждение" е равен на лог. 1, външното устройство не може да прави нов запис във входния регистър.
- Чрез процедура за сканиране или прекъсване, микропроцесорът чрез управляващия сигнал RD прочита съдържанието на входния регистър.

11

Програмираме паралелен интерфейс

- Интерфейсната схема установява сигнала "готовност" в лог.1, с което външното устройство се известява, че за него има записани данни.
- Външното устройство приема данните и връща сигнал "потвърждение" = лог.0.

8

Програмираме паралелен интерфейс

Процедурата при вход на данни е следната:

- Когато външното устройство има готови данни, които ще се въвеждат в микропроцесорната система, сигналът "готовност" се установява в лог.1 и данните се записват във входния регистър на интерфейса.

10

Програмираме паралелен интерфейс

- След прочитането на входния регистър, интерфейсът променя сигнал "потвърждение" в лог.0. С това процесът приключва и микропроцесорната система е готова да приема следващия байт.

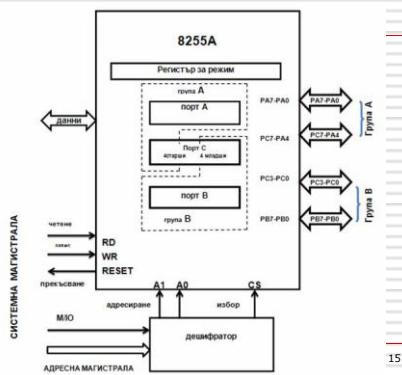
12

Програмираме паралелен интерфейс

В набора от интегрални микросхеми на микропроцесорна фамилия 8086 има специално разработена схема - 8255A, която изпълнява описаните функции.

13

Програмираме интерфейс 8255A



15

Програмираме паралелен интерфейс 8255A

Интегралната схема, блоковата схема на която е представена на фиг. 7.2, съдържа регистър за управление и три отделно адресируеми порта А, В и С.

14

Програмираме паралелен интерфейс 8255A

Порт С има специфични функции. Освен възможността за запис или четене на байт, всеки негов бит може да се установява поотделно в 0 или 1. Порт С обикновено се разглежда като съставен от две части - старша (битове C7 - C4) и младша (C3 - C0). Порт А и старшата част на порт С образуват група А, а порт В и младшата част на порт С образуват група В (фиг. 7.2).

17

Програмираме паралелен интерфейс 8255A

Взаимодействието на интерфейсната схема със системната магистрала се осъществява чрез двупосочните изводи за данни D7-D0, входовете CS, RD, WR за сигналите за управление и входовете A1, A0 за адресиране. За да се осъществи обмен с микропроцесора, сигнал CS трябва да е равен на лог.0, посоката на обмен (вход или изход) се определя от сигналите RD и WR, а състоянието на входовете A1, A0 определя частта от интерфейсната схема, с която ще се извърши обмена на данни. Адресирането на схемата е онагледено в табл. 7.1.

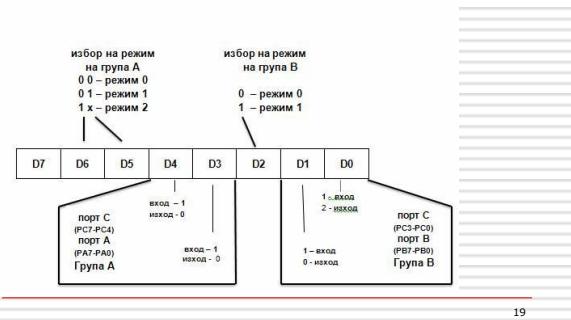
16

Програмираме паралелен интерфейс 8255A

Група А може да работи в режими 0, 1 и 2. Група В може да работи в режими 0 и 1. Режимите се определят от стойността на битовете на регистъра за режим. Предназначението им е показано на фиг. 7.3.

18

Предназначение на битовете в регистъра за режим



19

20

Програмираме паралелен интерфейс 8255А

Режим 1

В този режим се осъществява асинхронен обмен между интерфейса и устройствата за вход-изход. Портове А и В се използват за обмен на данни, а порт С - за сигнали за управление. Предназначението на отделните битове на порта С зависи от това, дали портовете А и В са програмирани за вход или за изход.

21

22

Програмираме паралелен интерфейс 8255А

■ **INTR** - използва се като заявка за прекъсване (активно ниво лог. 1). Останалите битове C7 и C6 от регистър С могат да се ползват като отделни линии за вход и изход.

23

24

Програмираме паралелен интерфейс 8255А

Режим 0

В този режим всеки от портовете може да бъде използван за паралелен вход или изход независимо от другите. Портовете могат да работят като вход и като изход, но не и като двупосочен вход-изход. Посоката се задава чрез битове D4, D3, D1 и D0, както е показано на фиг. 7.3.

Програмираме паралелен интерфейс 8255А

Ако и двата порта А и В са програмирани за вход, тогава предназначението на битовете от порт С е следното (табл. 7.3):

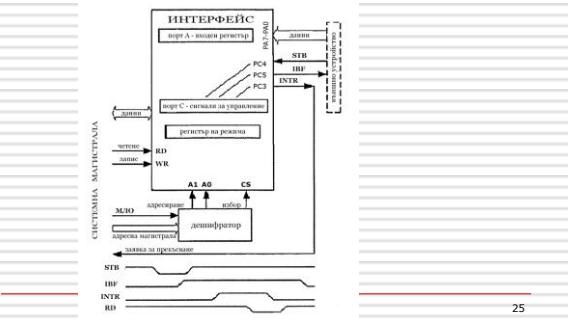
- **STB** - използва се за запис на данни от външното устройство в порта;
- **IBF** - показва готовността на порта да приема данни. При IBF = 1 в порта не може да се записва, защото има байт, който не е обработен;

Програмираме паралелен интерфейс 8255А

Процесът на вход на данни от външното устройство към микропроцесора в този режим е пояснен чрез блоковата схема и времедиаграмата от фиг. 7.4. Той протича в следната последователност (разглеждаме порт А):

- **Проверка за готовност на порт А да приеме байт данни.** Проверява се стойността на сигнал IBF. IBF = 1 означава, че във входния регистър на порт А има записани данни, които микропроцесорът не е успял, да прочете. При това положение външното устройство изчаква докато IBF стане лог.0.

Режим 1, работа на порт А при вход на данни



25

Програмираме паралелен интерфейс 8255А

- Вход на данни в порт А. При IBF = 0 външното устройство извежда байт данни към входовете PA7 – PA0 на порт А и подава отрицателен импулс на вход STRB. Чрез STRB се извършва запис на данните във входния регистър. С извършването на записи изход IBF също се установява в лог.1, с което се блокира възможността за нов запис. Изход INTR също се установява в лог.1, с което към микропроцесора се подава заявка за прекъсване.

26

Програмираме паралелен интерфейс 8255А

- Обработка на заявката за прекъсване. Микропроцесорът прекъсва изпълнението на текущата програма (ако това е разрешено) и прочита байта от порт А чрез управляващия сигнал за четене RD. Същият сигнал RD нулира изходи IBF и INTR. С това порт А е готов за нова входна операция.

Описаният процес на вход на данни важи и за порт В.

27

Програмираме паралелен интерфейс 8255А

Ако и двата порта А и В са програмирани за изход на данни, тогава предназначението на битовете от порт С е следното (табл. 7.4).

28

Програмираме паралелен интерфейс 8255А

Процесът на изход на данни от микропроцесора към външното устройство в този режим е пояснен чрез блоковата схема и времедиаграмата от фиг. 7.5 и протича в следната последователност (разглеждаме порт А):

29

Програмираме паралелен интерфейс 8255А

- Изходно състояние. Сигнали OBF, ACK и INTR са в състояние лог.1;
- Запис на байт в порт А. Микропроцесорът записва байт данни в порт А чрез управляващия сигнал за запис WR. Същият сигнал WR нулира изход INTR;

30

Програмираме паралелен интерфейс 8255А

- Изход на данни от порт A. Интерфейсът подава сигнал OBF = лог.0 към външното устройство, с което го известява, че има данни за него. След като приеме данните, външното устройство връща за потвърждение сигнал ACK=0.
- Завършване на цикъла. С приемането на сигнал ACK интерфейсът превключва сигнали OBF и INTR в лог.1, с което процесът завършва и порт A се връща в изходно състояние.

31

Програмираме паралелен интерфейс 8255А

Промяната на сигнал INTR от 0 може да се използва за заявка за прекъсване към микропроцесора и инициализиране на нов цикъл за изход на данни.

32

Програмираме паралелен интерфейс 8255А

□ Режим 2

В този режим може да работи само порт A, който става двупосочен. В същото време порт B може да бъде програмиран за работа в режим 0 или 1. Обозначението на битовете от порт C, е представено в таблица 7.5. Показани са само битовете, отнасящи се до порт A. За порт B те зависят от режима и са показани в предходните таблици.

33

Програмираме паралелен интерфейс 8255А

Действието в режим 2 е пояснено чрез блоковата схема и времедиаграмата от фиг. 7.6. То е аналогично на предходния режим 1 и зависи от това каква операция - за вход или за изход се изпълнява. Едновременната работа на порт A и в двете посоки е възможна, защото има два отделни регистъра за вход и за изход, които се ползват независимо един от друг чрез сигналите за управление.

34

Край на част 8

35