

Chương 3

GIAO TIẾP QUA CỎNG SONG SONG

CỔNG SONG SONG



Parallel PCMCIA Card SPP/ECP/EPP LPT Port True Printer Port CardBus

CỒNG SONG SONG

Ưu điểm

- Lập trình đơn giản, dễ kết nối
- Tốc độ nhanh

Nhược điểm

- Khoảng cách nhỏ
- Chống nhiễu kém

CỔNG SONG SONG

Cổng song song có 5 chế độ hoạt động, tùy loại mainboard, PC có thể hỗ trợ đủ 5 mode hoặc vài mode

SPP: Cổng song song chuẩn

- Compatibility: Xuất 8 bit
- Nibble: nhập 4 bit
- Byte: Cổng 2 chiều

EPP: Cổng tăng cường

- Enhanced Parallel Port

ECP: Cổng song song mở rộng

- Extended Capability Port

CỔNG SONG SONG

Cổng SPP có thể truyền dữ liệu song song 8 bit từ máy tính ra với vận tốc 50 Kbytes/sec đến 150 Kbytes/sec. Khi muốn nhập dữ liệu vào máy tính có thể dùng mode Nibble truyền 4 bit hay mode Byte truyền 8 bit.

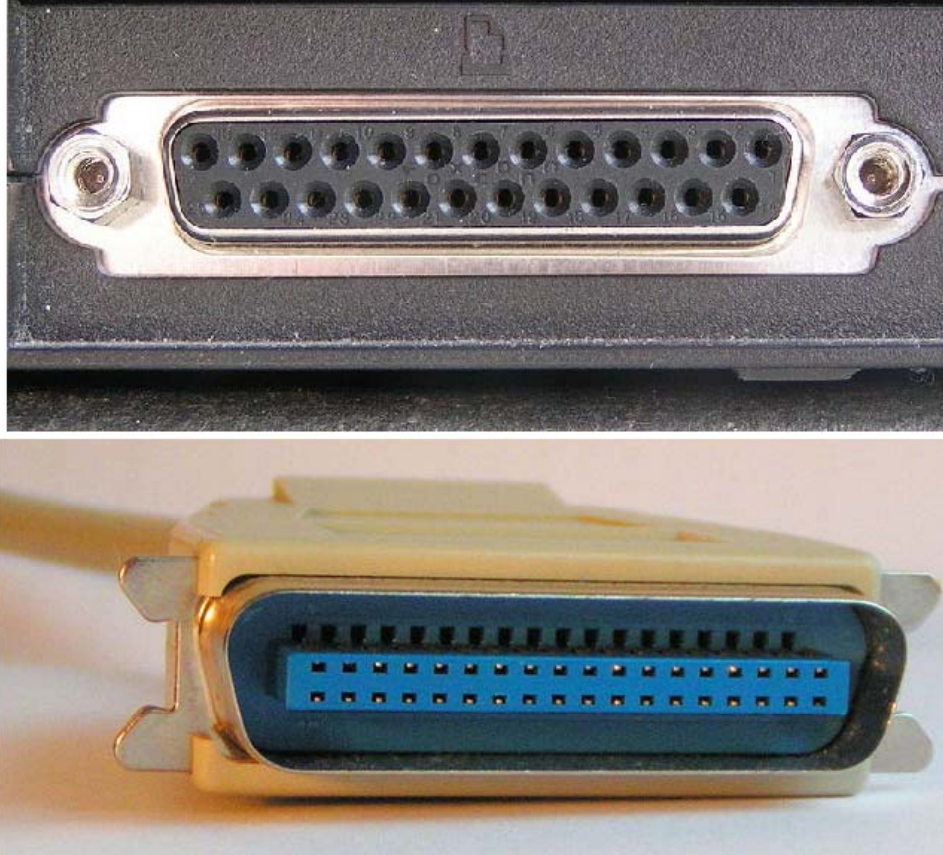
Cổng EPP và ECP dùng thêm phần cứng hỗ trợ nên vận tốc truyền nhanh hơn, có thể đến 2 Mbytes/sec, thu phát song song 8 bit.

Thay đổi chế độ của cổng song song bằng cách vào BIOS SETUP của máy tính khi khởi động máy.

Các máy tính không có cổng song song có thể sử dụng chuyển đổi USB - Parallel Port hay PCMCIA (Personal Computer Memory Card International Association) - *Parallel Port*.

CỔNG SONG SONG

Cổng SSP



Có đầu nối 25 chân cái thường dùng để kết nối với máy in đầu nối Centronics 34 chân.

CỔNG SONG SONG

Cổng SSP

Chân D-25	Centronics	Tín hiệu SSP	Hướng	Thanh ghi	Đảo
1	1	/Strobe	Vào/Ra	Điều khiển	Có
2	2	Data 0	Ra	Data	
3	3	Data 1	Ra	Data	
4	4	Data 2	Ra	Data	
5	5	Data 3	Ra	Data	
6	6	Data 4	Ra	Data	
7	7	Data 5	Ra	Data	
8	8	Data 6	Ra	Data	
9	9	Data 7	Ra	Data	

Ngõ ra cổng song song tương thích với cổng TTL, dòng cấp và thu khoảng vài *mA* đến 16mA tùy loại kết cấu phần cứng (công nghệ ASIC)

CỔNG SONG SONG

Cổng SSP

Chân D-25	Centronics	Tín hiệu SSP	Hướng	Thanh ghi	Đảo
10	10	/Ack	Vào	Điều khiển	
11	11	Busy	Vào	Trạng thái	Có
12	12	Paper Out	Vào	Trạng thái	
13	13	Select	Vào	Trạng thái	
14	14	/Auto-Linefeed	Vào/Ra	Điều khiển	Có
15	32	/Error	Vào	Trạng thái	
16	31	/Init	Vào/Ra	Điều khiển	
17	36	/Select Printer	Vào/Ra	Điều khiển	Có
18-25	19-30	Mass			

CỔNG SONG SONG

Cổng SSP

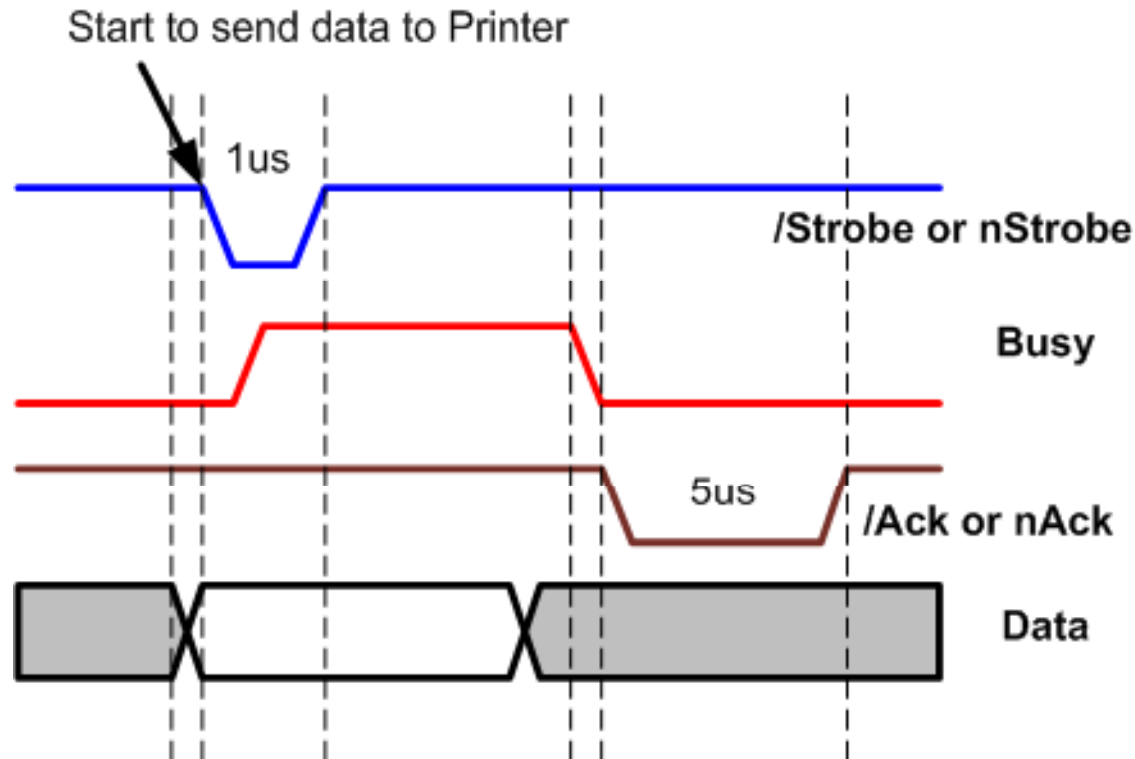
At Standard TTL Levels

Signal Name		Adapter Pin Number	
← - Strobe		1	
E	← +Data Bit 0	2	P
X	← +Data Bit 1	3	A
T	← +Data Bit 2	4	R
E	← +Data Bit 3	5	A
R	← +Data Bit 4	6	L
N	← +Data Bit 5	7	L
A	← +Data Bit 6	8	E
L	← +Data Bit 7	9	L
- Acknowledge		10	→
D	+Busy	11	→ A
E	+Paper End	12	→ D
V	+Select	13	→ A
I	← -Auto Feed	14	P
C	← -Error	15	→ T
E	← -Initialize	16	E
← -Select Input		17	R
Ground		18-25	

Kết nối giữa cổng SSP và ngoại vi

CỔNG SONG SONG

Cổng SSP



Bắt tay dữ liệu

Ban đầu dữ liệu 8 bit được đưa ra,
Nếu chân Busy ở mức thấp thì máy in rảnh,
Nếu Busy sẽ ở mức cao. Thì máy in bận
Tín hiệu Strobe rộng 1μsec và máy in đọc dữ liệu,
Máy in đọc xong sẽ báo trở lại bằng nACK rộng 5μsec.

CỔNG SONG SONG

Cổng SSP

Việc giao tiếp được thực hiện qua 3 thanh ghi: thanh ghi dữ liệu, thanh ghi trạng thái và thanh ghi điều khiển. Thông thường sử dụng hai địa chỉ gốc là: **378 cho LPT1 và 278 cho LPT2**

Một số máy dùng địa chỉ 03BC.

- Thanh ghi dữ liệu có địa chỉ gốc +0, 8 bit, nhận dữ liệu để xuất ra ngoài, dữ liệu được chốt..
- Thanh ghi trạng thái địa chỉ gốc +1 là thanh ghi chỉ đọc dùng để nhận tín hiệu từ ngoài vào, có 5 tín hiệu vào.
- Thanh ghi điều khiển có bốn đường ra điều khiển, địa chỉ gốc +2, các đường này dùng cực thu hở do đó có thể giao tiếp hai chiều.
- Các chân 1, 11, 14 và 17 được đảo phân cực, bit D6 thanh ghi trạng thái (chân số 10) từ 10 thì gây ra ngắt IRQ7 nếu được cho phép bởi D4 của thanh ghi điều khiển =1.

CỔNG SONG SONG

Cổng SSP

Thanh ghi trạng thái

Bit	D7	D6	...	...	...	...		D0
Function	Busy	/ACK	Paper Out	Select	/Error	/IRQ	X	X
Pin	11	10	12	13	15			

Chú ý là bit Busy được đảo, nghĩa là nếu chân 11 có điện áp +5V thì bit D7 của thanh ghi trạng thái mức logic 0, bit D2 bằng 0 khi có ngắt từ /ACK.

Thanh ghi Điều khiển

Bit	D7	D6	...	...	...	...		D0
Function	X	X	Cho phép cổng 2 chiều	Cho phép ngắt qua ACK	Chon máy in /Select	Khởi động máy in /Init	Xuống hàng /Auto Linefeed	Kích /Strobe
Pin					17	16	14	1

CỔNG SONG SONG

Cổng SSP

	7	6	5	4	3	2	1	0	
DATA	X	X	X	X	X	X	X	X	Base = 278/378/3BC H
STATUS	/X	X	X	X	X	-	-	-	Base +1
CONTROL	-	-	-	-	/X	X	/X	/X	Base +2

Địa chỉ của port song song

S7,C0,C1,C3 bị đảo

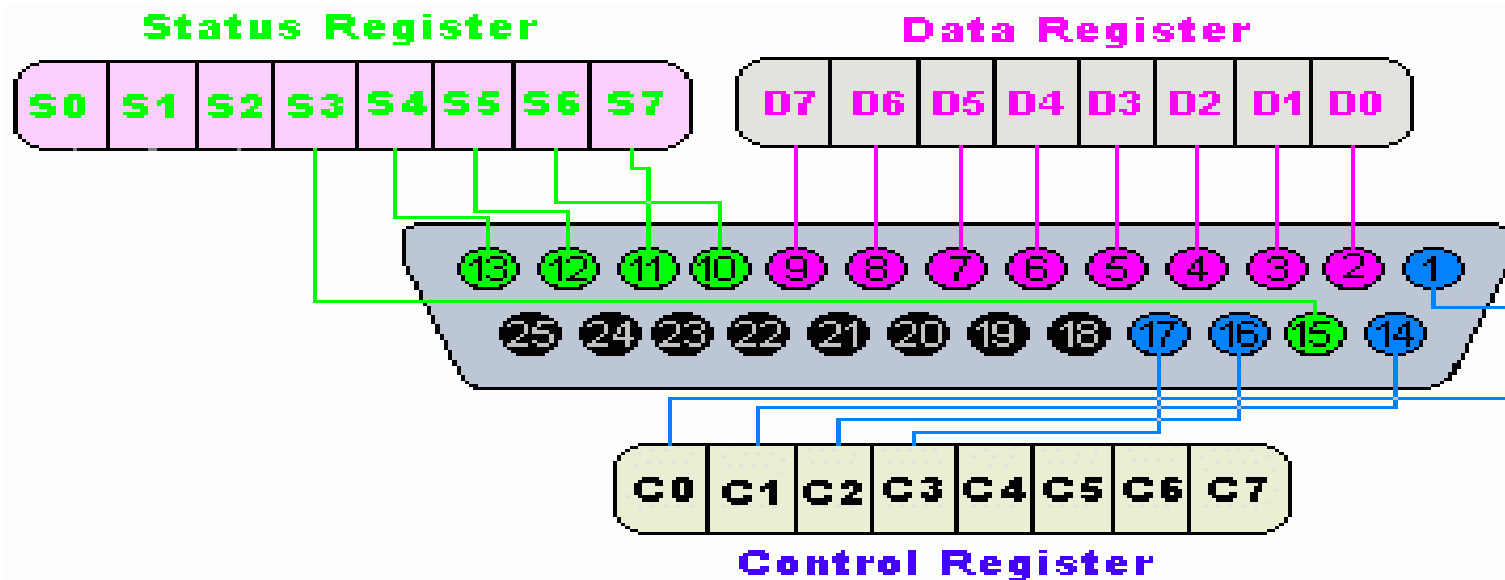
Nếu chân 11 ở mức cao thì $S7 = 0$

$C0 = 1$ gây cho chân 1 của cổng song song ở mức thấp

CỔNG SONG SONG

Cổng SSP

Address	Notes:
3BCh - 3BFh	Used for Parallel Ports which were incorporated on to Video Cards - Doesn't support ECP addresses
378h - 37Fh	Usual Address For LPT 1
278h - 27Fh	Usual Address For LPT 2



LẬP TRÌNH Giao tiếp CỒNG SONG SONG

Chương trình giao tiếp trên VB sử dụng thư viện liên kết động để trao đổi dữ liệu với cổng máy in.

Có các file liên kết động sau: IO.DLL; inpout32.dll → tìm trên mạng

Copy những file này vào thư mục system32 của máy tính

Khai báo các hàm cần được sử dụng trong chương trình giao tiếp.

LẬP TRÌNH Giao tiếp CỔNG SONG SONG

Thư viện IO.DLL bao gồm các hàm sau:

-Hàm PortOut: xuất 1 byte ra cổng

*Private Declare Sub PortOut Lib "IO.DLL" (ByVal Port As Integer,
ByVal Data As Byte)*

Port: địa chỉ của cổng

Data: dữ liệu xuất

LẬP TRÌNH Giao tiếp CỔNG SONG SONG

Thư viện IO.DLL bao gồm các hàm sau:

-Hàm PortWordOut: xuất 1 word ra cổng

*Private Declare Sub PortWordOut Lib "IO.DLL" (ByVal Port As Integer,
ByVal Data As Integer)*

Port: địa chỉ của cổng

Data: dữ liệu xuất

LẬP TRÌNH Giao tiếp CỔNG SONG SONG

Thư viện IO.DLL bao gồm các hàm sau:

Hàm PortDWordOut: xuất 1 double word ra cổng

Private Declare Sub PortDWordOut Lib "IO.DLL" (ByVal Port As Integer, ByVal Data As Long)

Port: địa chỉ của cổng

Data: dữ liệu xuất

LẬP TRÌNH Giao tiếp CỔNG SONG SONG

Thư viện IO.DLL bao gồm các hàm sau:

-Hàm PortIn: nhập 1 byte từ cổng, trả về giá trị nhập

Private Declare Function PortIn Lib "IO.DLL" (ByVal Port As Integer) As Byte

- Hàm PortWordIn: nhập 1 word từ cổng

Private Declare Function PortWordIn Lib "IO.DLL" (ByVal Port As Integer) As Integer

- Hàm PortDWordIn: nhập 1 double word từ cổng

Private Declare Function PortDWordIn Lib "IO.DLL" (ByVal Port As Integer) As Long

LẬP TRÌNH Giao tiếp CỒNG SONG SONG

Thư viện Inpout.DLL bao gồm các hàm sau:

short Inp32(short PortAddress)

Đọc data từ Port

Private Declare Function Inp32 Lib "Inpout.DLL" (ByVal Port As Integer) As Integer

void Out32(short PortAddress, short data)

Xuất data ra Port

Private Declare Function Out32 Lib "Inpout.DLL" (ByVal Port As Integer, ByVal Data As Integer)

LẬP TRÌNH Giao tiếp CỒNG SONG SONG

Ngoài ra còn có Thư viện hwinterface32.DLL bao gồm các hàm sau:

Ngoài thư viện của Inpout.dll, còn có các thư viện khác như của IO.dll

short Inp32(short PortAddress)

Đọc data từ Port

Private Declare Function Inp32 Lib "Inpout.DLL" (ByVal Port As Integer) As Integer

void Out32(short PortAddress, short data)

Xuất data ra Port

Private Declare Function Out32 Lib "Inpout.DLL" (ByVal Port As Integer, ByVal Data As Integer)

LẬP TRÌNH Giao tiếp CỒNG SONG SONG

Ngoài ra còn có Thư viện hwinterface32.DLL bao gồm các hàm sau:

Function	Description
short InpW32(short PortAddress)	Reads a Word from the specified I/O address.
void OutW32(short PortAddress, short data)	Writes a Word to the specified I/O address.
long InpD32(short PortAddress)	Reads a DWord from the specified I/O address.
void OutD32(short PortAddress, long data)	Writes a DWord to the specified I/O address.
long ReadMemChar(ULONG Address)	Reads a single Byte to the specified address
long ReadMemShort(ULONG Address)	Reads a 16 bit word to the specified address
long ReadMemLong(ULONG Address)	Reads a 32 bit Dword to the specified address
void WriteMemChar(ULONG Address, UCHAR data)	Writes a single Byte to the specified address
void WriteMemShort(ULONG Address, USHORT data)	Writes a single Word to the specified address
void WriteMemLong(ULONG Address, ULONG data)	Writes a single DWord to the specified address.

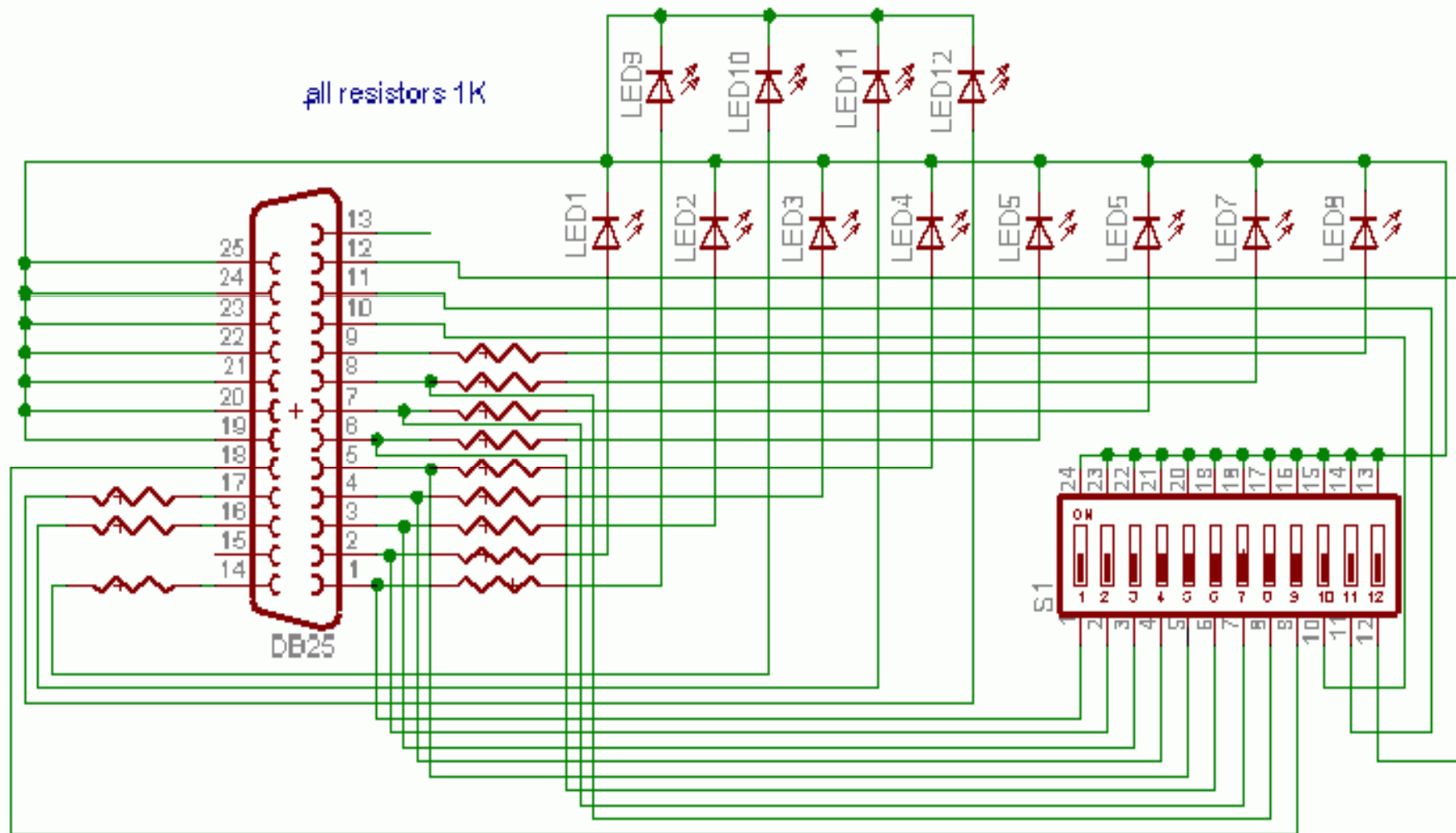
LẬP TRÌNH Giao tiếp CỒNG SONG SONG

Một số địa chỉ cổng tiêu biểu

Address	Port
400H	Base I/O address for serial port 1 (COM 1)
402H	Base I/O address for serial port 2 (COM 2)
404H	Base I/O address for serial port 3 (COM 3)
406H	Base I/O address for serial port 4 (COM 4)
408H	Base I/O address for parallel port 1 (LPT 1)
40AH	Base I/O address for parallel port 2 (LPT 2)
40CH	Base I/O address for parallel port 3 (LPT 3)

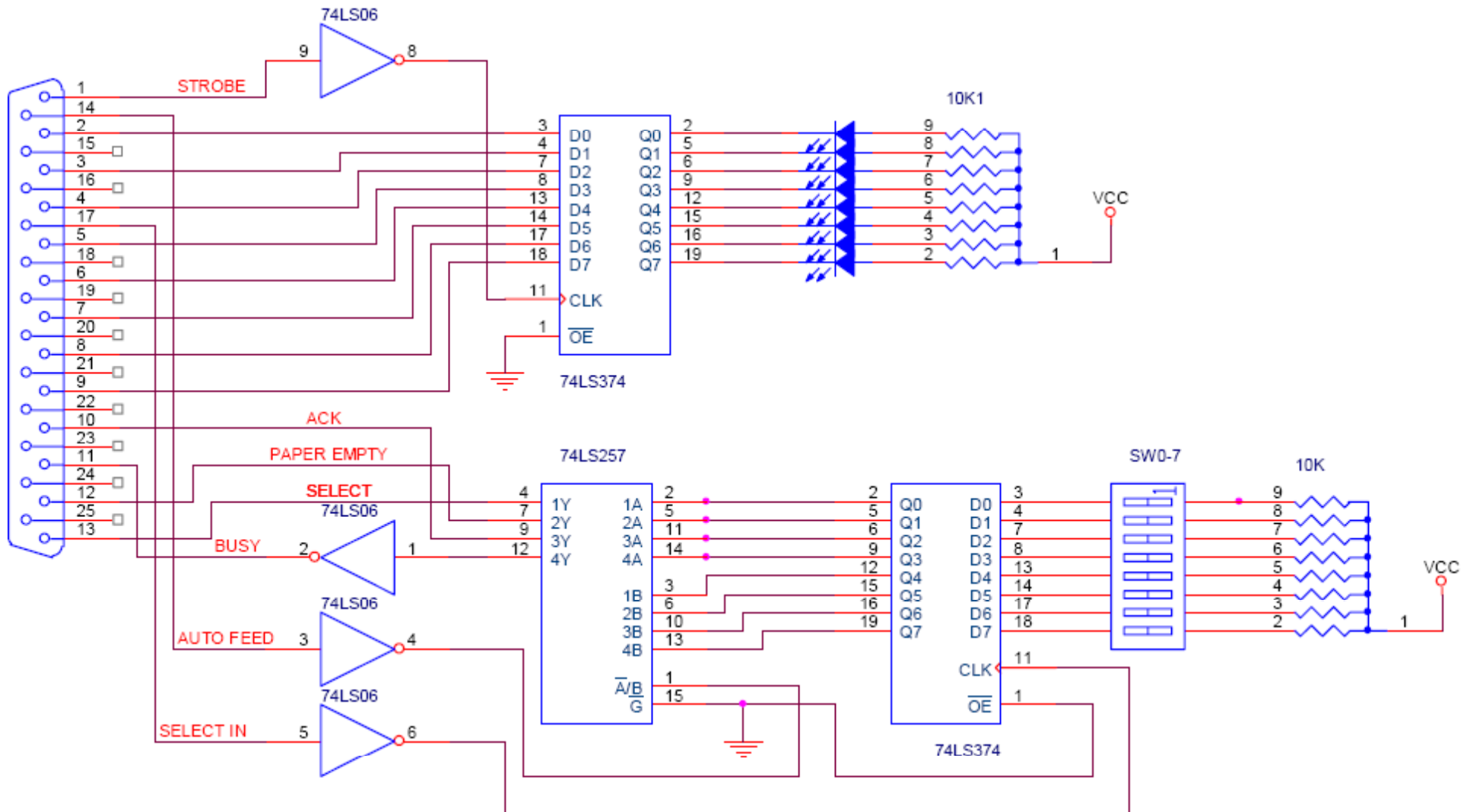
LẬP TRÌNH Giao tiếp CỒNG SONG SONG

Giao tiếp đơn giản, chế độ SSP bi-directional mode



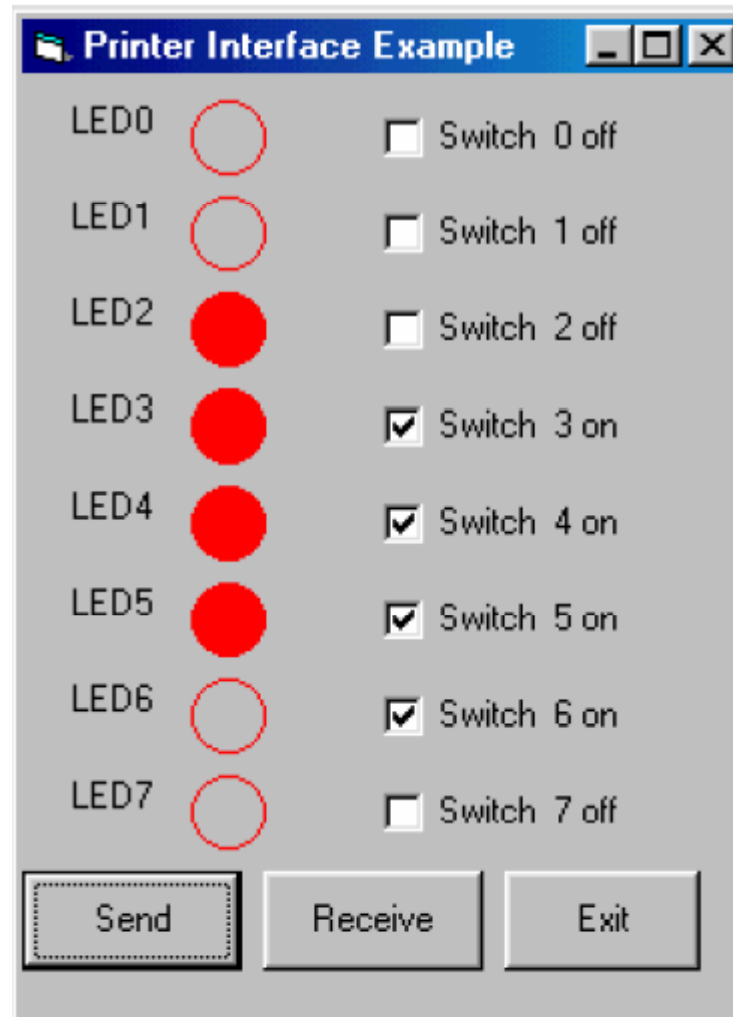
LẬP TRÌNH Giao tiếp CỒNG SONG SONG

Giao tiếp ngoại vi với SPP kết hợp đọc dữ liệu bằng status byte:



LẬP TRÌNH Giao tiếp CỒNG SONG SONG

Giao diện VB:



LẬP TRÌNH Giao tiếp CỔNG SONG SONG

Chương trình:

‘Khai báo hàm cho IO.dll

Private Declare Sub **PortOut** Lib "IO.DLL" (ByVal Port As Integer,
ByVal Data As Byte)

Private Declare Function **PortIn** Lib "IO.DLL" (ByVal Port As
Integer) As Byte

‘Khai báo biến địa chỉ port LPT

Private **BA_LPT** As Integer

LẬP TRÌNH Giao tiếp CỒNG SONG SONG

Chương trình:

‘Hàm cho button Receive

Private Sub **cmdReceive_Click()**

Dim n As Integer

Dim n1 As Integer

Dim i As Integer

PortOut BA_LPT + 2, &H8 'SELECTIN = 1

PortOut BA_LPT + 2, 0 'SELECTIN = 0

n1 = PortIn(BA_LPT + 1) 'Doc 4 bit thap

n1 = n1 / &H10 'Dich phai 4 bit

PortOut BA_LPT + 2, 2 'AUTOFEED=1

n = PortIn(BA_LPT + 1) 'Doc 4 bit cao

n = n And &HF0

n = n + n1

LẬP TRÌNH Giao tiếp CỒNG SONG SONG

Chương trình:

‘Hàm cho button Receive

For i = 0 To 7

chkSW(i).Value = n Mod 2

If chkSW(i).Value = 0 Then chkSW(i).Caption = "Switch " & Str(i) &
off"

Else chkSW(i).Caption = "Switch " & Str(i) & " on"

End If

n = Fix(n / 2)

Next i

End Sub

LẬP TRÌNH Giao tiếp CỔNG SONG SONG

Chương trình:

‘Hàm cho button Send

Private Sub **cmdSend_Click()**

Dim t As Integer

Dim i As Integer

Dim s As String

t = 0

For i = 0 To 7

t = t + (2 ^ i) * (1 - shpLED(i).FillStyle)

Next i

PortOut BA_LPT, t

PortOut BA_LPT+2, 1 'STROBE = 1

PortOut BA_LPT+2, 0 'STROBE = 0

End Sub

LẬP TRÌNH Giao tiếp CỒNG SONG SONG

Chương trình:

‘Hàm cho

```
Private Sub Form_Load()
```

```
BA_LPT = &H378
```

```
PortOut BA_LPT + 2, 0
```

```
End Sub
```

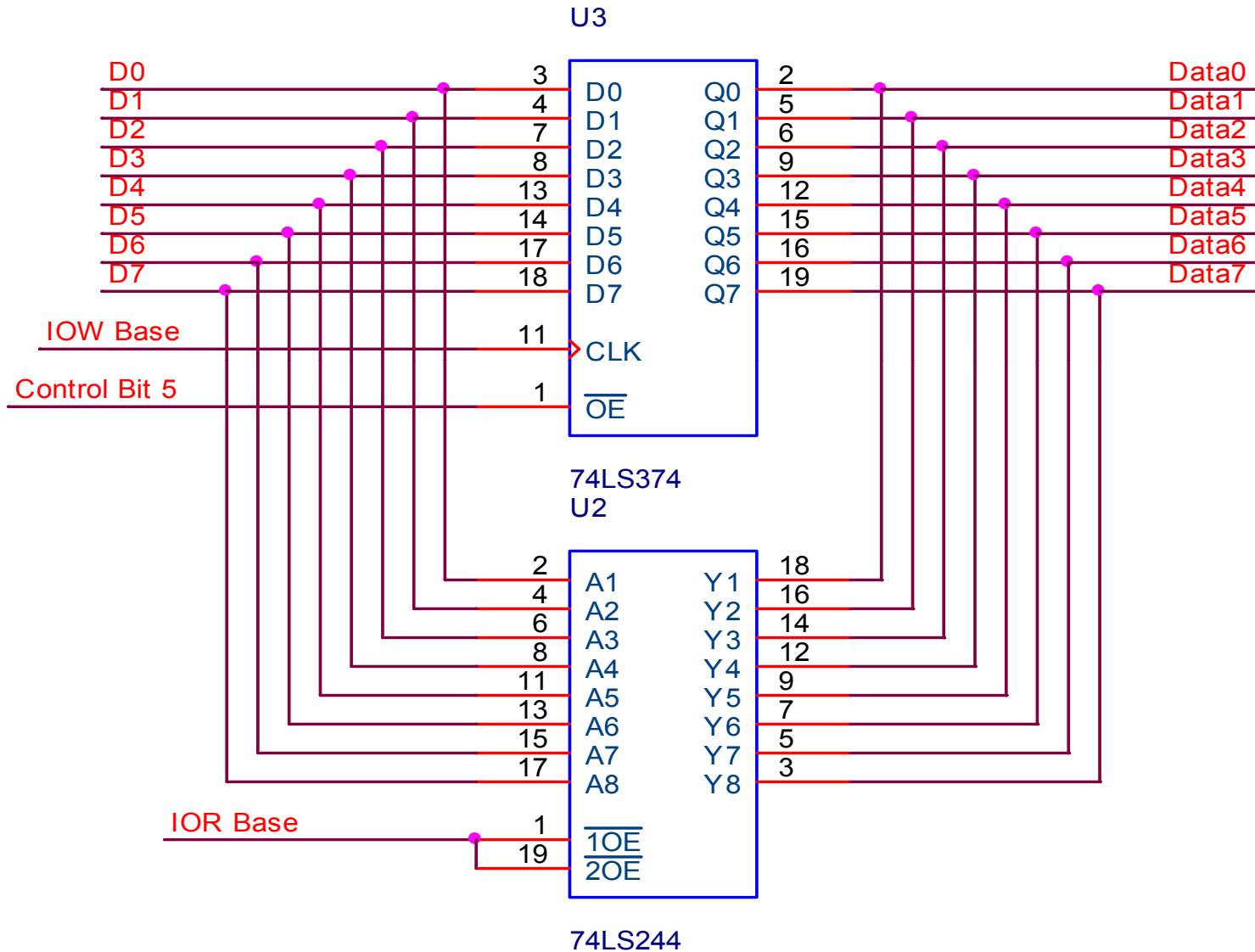
```
Private Sub lblLED_Click(Index As Integer)
```

```
shpLED(Index).FillStyle = 1 - shpLED(Index).FillStyle
```

```
End Sub
```

MỞ RỘNG CỔNG SONG SONG

Xuất nhập dữ liệu 8 bit chế độ Bidirectional

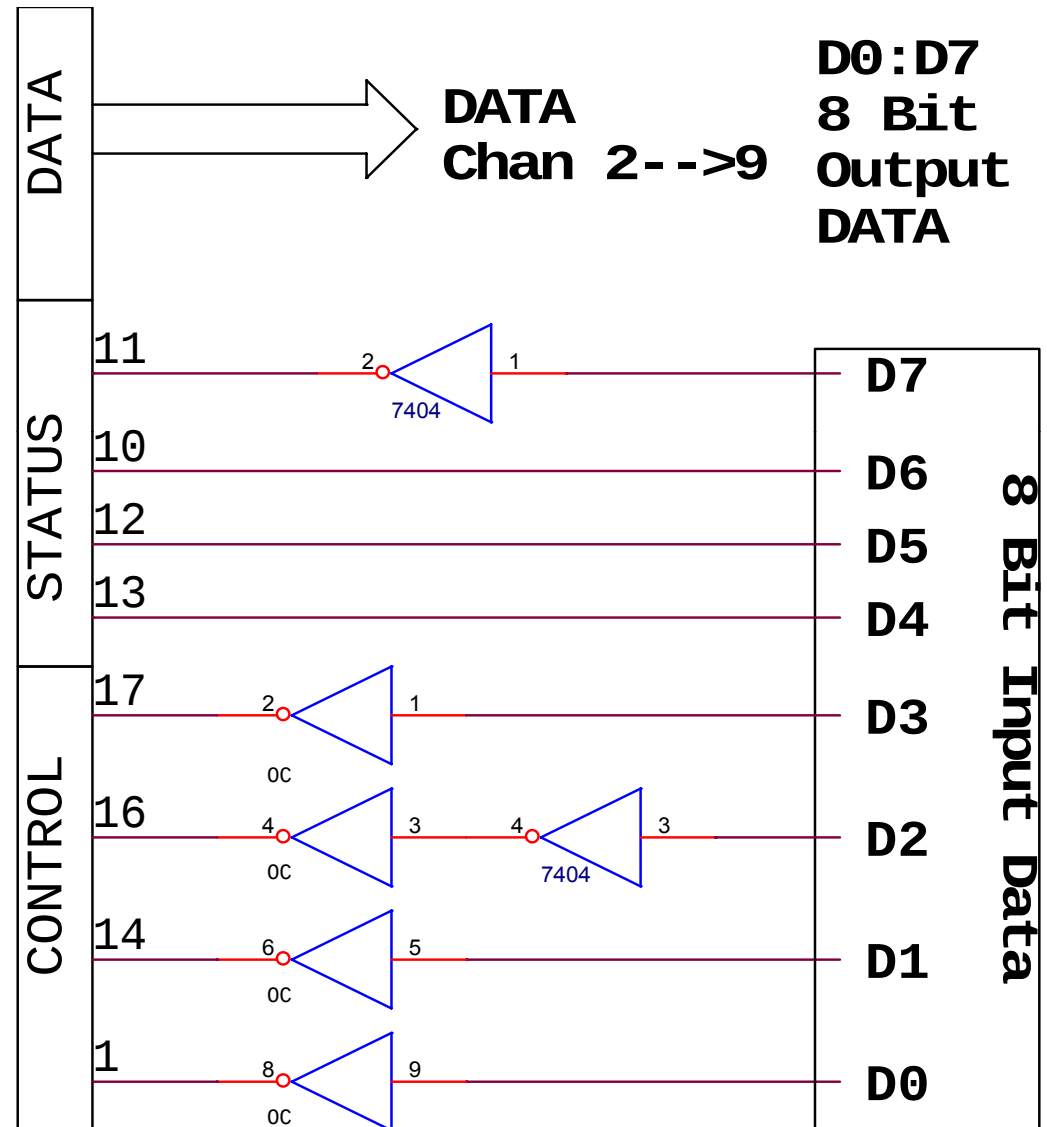


MỞ RỘNG CỔNG SONG SONG

Xuất nhập dữ liệu 8 bit chế độ Bidirectional

Bit 5 của port điều khiển cho phép hoặc cấm chức năng 2 chiều của Cổng song song. Chức năng này chỉ thực hiện được đối với những cổng có hỗ trợ. Khi bit này được set lên 1, chân 2→9 sẽ chuyển sang dạng tổng trở cao. Ở trạng thái này, ta có thể nhận dữ liệu từ data port. Việc ghi dữ liệu vào data port sẽ được lưu lại nhưng không được thực hiện. Tắt chế độ 2 chiều bằng cách set bit 5 xuống mức 0.

Đọc dữ liệu 8 bit với cổng SSP 1 chiều dùng Status và Control Byte



MỞ RỘNG CỔNG SONG SONG

Đọc dữ liệu 8 bit với cổng SSP 1 chiều dùng Status và Control Byte

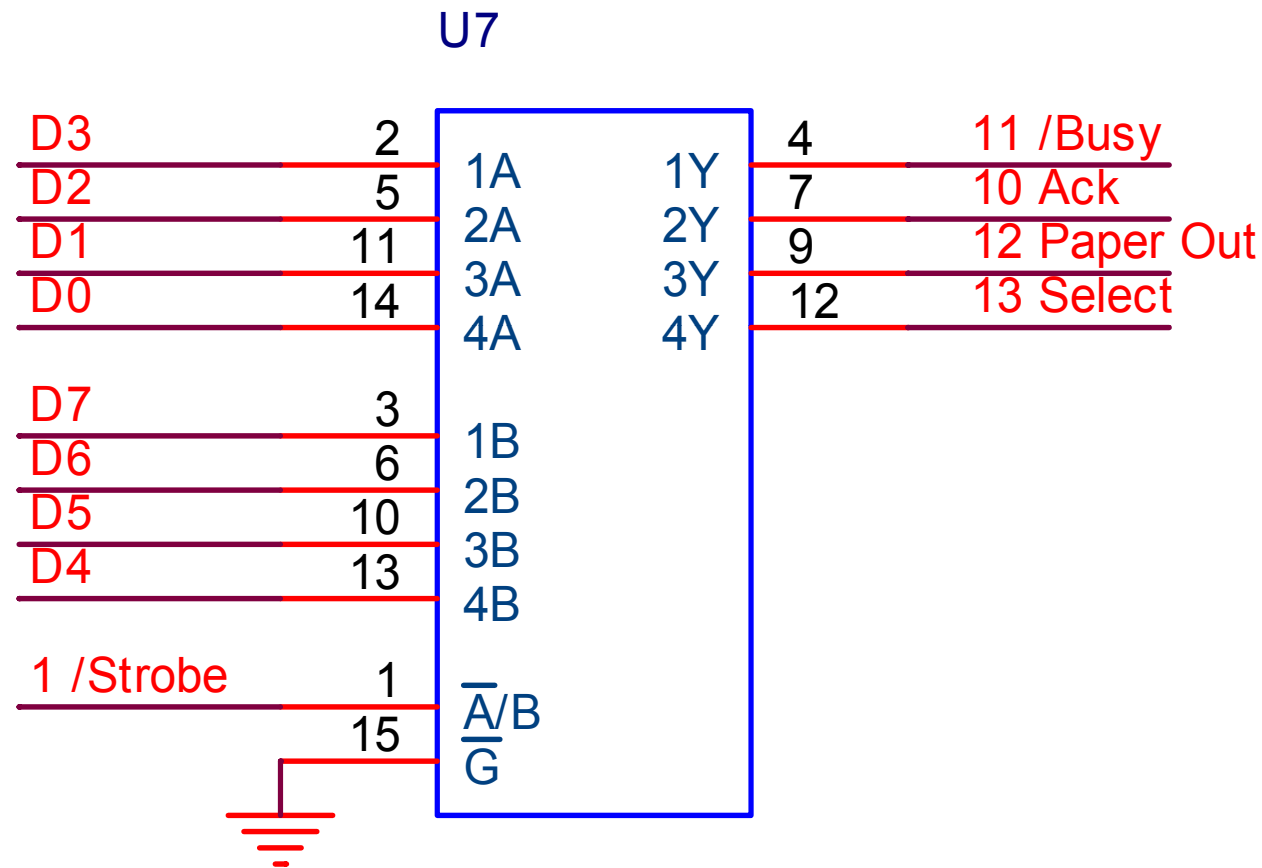
Port điều khiển dùng để đọc 4 bit thấp. Như đã đề cập ở trên, port điều khiển có các ngõ ra cực thu hở, có 2 trạng thái là tổng trở cao và GND. Nếu ta kết nối các ngõ vào trực tiếp vào port, sẽ có xung đột xảy ra nếu ngõ vào ở mức cao và port đang kéo xuống. Vì vậy ta thường sử dụng các bộ đảo cực thu hở.

Đọc dữ liệu

```
outportb(CONTROL, inportb(CONTROL) & 0xF0 | 0x04);  
a = (inportb(STATUS) & 0xF0); /* Read MSnibble */  
a = a |(inportb(CONTROL) & 0x0F); /* Read LSnibble */  
a = a ^ 0x84; /* Toggle Bit 2 & 7 */
```

MỞ RỘNG CỔNG SONG SONG

Đọc dữ liệu 8 bit với cổng SSP 1 chiều dùng Status và bộ phân kênh,
Chế độ Nibble



74LS157

8 Inputs Using 74LS157 Multiplexer

MỞ RỘNG CỔNG SONG SONG

Đọc dữ liệu 8 bit với cổng SSP 1 chiều dùng Status và bộ phân kênh,
Chế độ Nibble

Chế độ Nibble là chế độ được yêu thích khi đọc 8 bit dữ liệu mà không đặt port trong chế độ ngược lại sử dụng các đường dữ liệu. Chế độ Nibble sử dụng bộ phân kênh $4 \times 2 \rightarrow 4 \times 1$ để đọc dữ liệu. Software sau đó sẽ kết hợp 2 nibble thành byte. Một khuyết điểm của chế độ này là tốc độ thấp hơn. Chế độ này yêu cầu nhiều lệnh I/O hơn để đọc 1 byte, nó đòi hỏi phải sử dụng IC ngoài.

MỞ RỘNG CÔNG SONG SONG

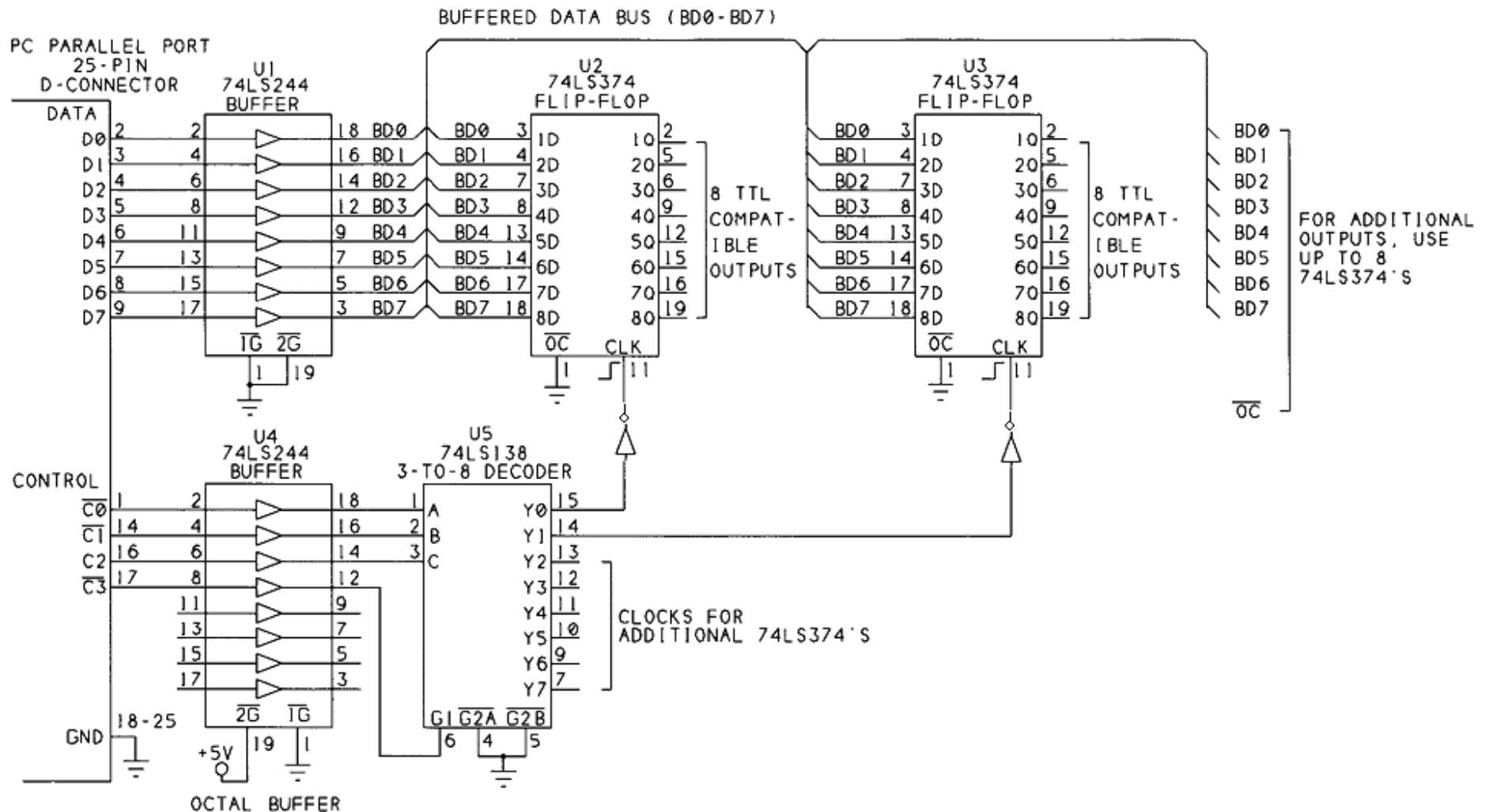
Đọc dữ liệu 8 bit với cổng SSP 1 chiều dùng Status và bộ phân kênh,
Chế độ Nibble

Đọc dữ liệu với chế độ Nibble

```
outportb(CONTROL, inportb(CONTROL) | 0x01); /* Select Low Nibble (A)*/  
a = (inportb(STATUS) & 0xF0); /* Read Low Nibble */  
a = a >> 4; /* Shift Right 4 Bits */  
outportb(CONTROL, inportb(CONTROL) & 0xFE); /* Select High Nibble (B)*/  
a = a |(inportb(STATUS) & 0xF0); /* Read High Nibble */  
byte = byte ^ 0x88;
```

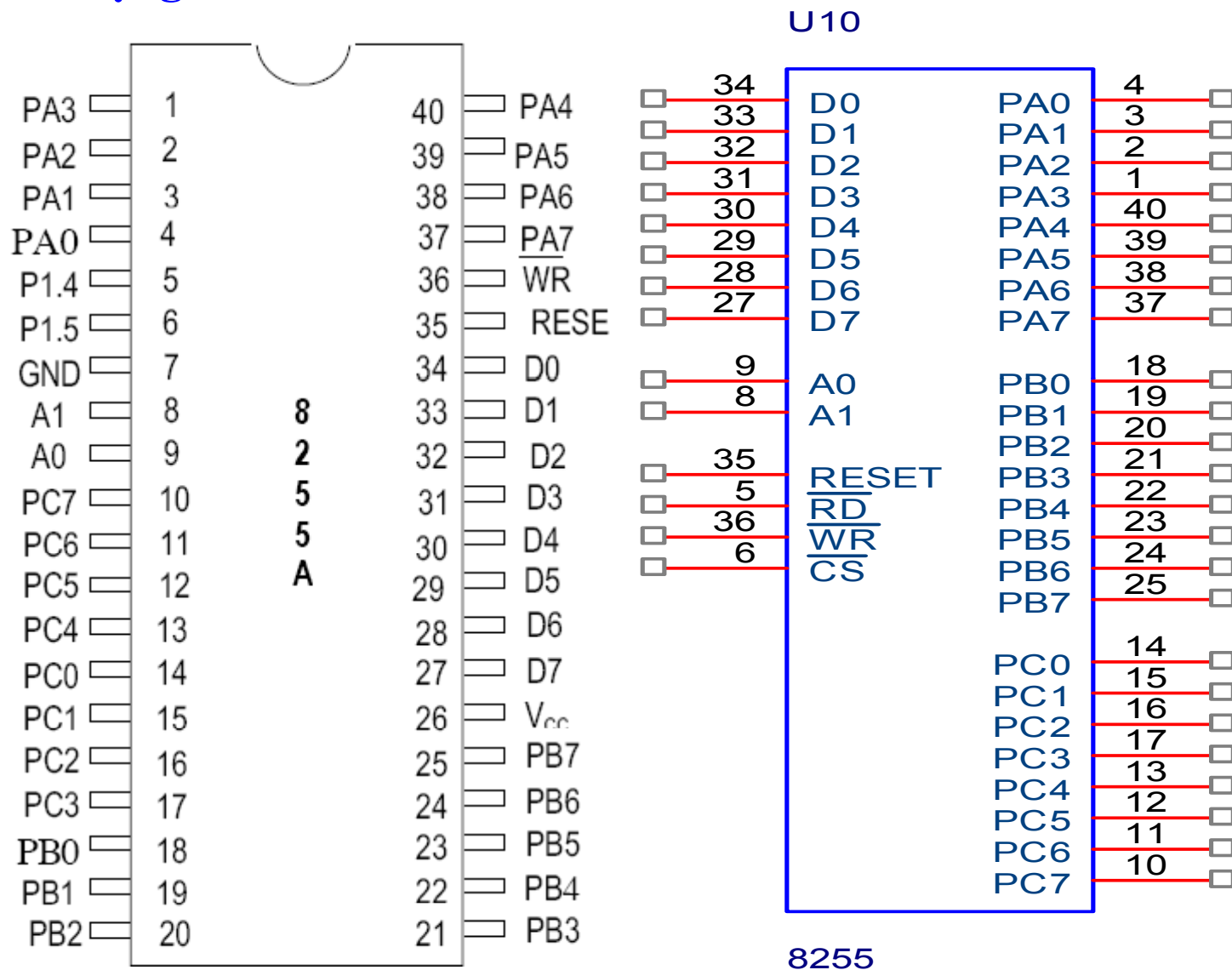
MỞ RỘNG CỔNG SONG SONG

Mở rộng ngõ ra với 74LS374



MỞ RỘNG CỔNG SONG SONG

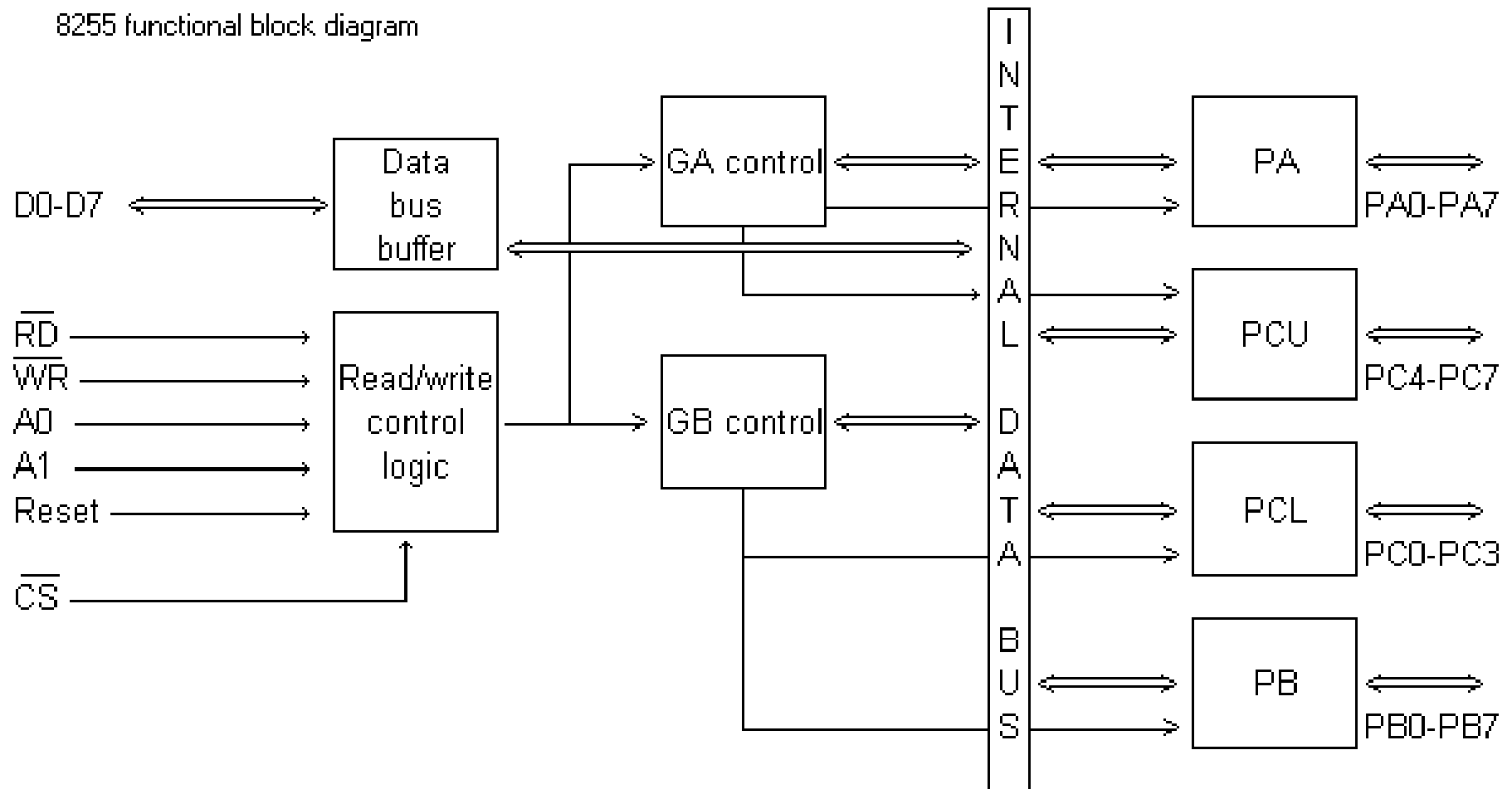
Mở rộng Port với IC 8255



MỞ RỘNG CỔNG SONG SONG

Mở rộng Port với IC 8255

8255 functional block diagram



MỞ RỘNG CÔNG SONG SONG

Mở rộng Port với IC 8255

8 bit của Port A có thể được lập trình như 8 bit đầu vào , 8 bit đầu ra, hoặc vào/ra

8 bit của Port B có thể được lập trình như 8 bit đầu vào , 8 bit đầu ra, hoặc vào/ra

8 bit của Port C có thể được lập trình như 8 bit đầu vào , 8 bit đầu ra. Có thể chia làm 2 phần: digit cao = 4 bit cao, và digit thấp = 4 bit thấp. Mỗi digit có thể lập trình như các đầu vào hoặc các đầu ra. Ngoài ra, các bit trên Port C cũng có thể được lập trình riêng rẽ

MỞ RỘNG CỔNG SONG SONG

Mở rộng Port với IC 8255

/WR và /RD là 2 tín hiệu tích cực mức thấp dùng để ghi và đọc dữ liệu.

D0 –D7 : các chân dữ liệu giao tiếp VXL hay cổng

RESET tích cực mức cao, dùng để xóa thanh ghi điều khiển. Tất cả các cổng được reset thành các cổng vào.

/CS tích cực thấp dùng để chọn chip

A0,A1: dùng để truy cập các cổng A, B, C và thanh ghi điều khiển.

MỞ RỘNG CỔNG SONG SONG

Các chế độ hoạt động IC 8255

Mode 0: Các Port A, B, CU, CD chỉ có thể được lập trình như các Port vào hoặc Port ra đơn giản.

Mode1: Port A, B được lập trình như các cổng vào ra có khả năng bắt tay. Tín hiệu bắt tay được cấp bởi các bit ở cổng C.

Mode 2: Cổng A được dùng như cổng vào/ra 2 chiều có khả năng bắt tay, với các tín hiệu bắt tay được cấp bởi các bit ở cổng C. Cổng B được lập trình như các Port vào hoặc Port ra đơn giản hoặc có bắt tay

Mode BRS: Chế độ thiết lập hoặc xóa bit. Chế độ này chỉ có những bit riêng rẽ của cổng C có thể được lập trình.

MỞ RỘNG CỔNG SONG SONG

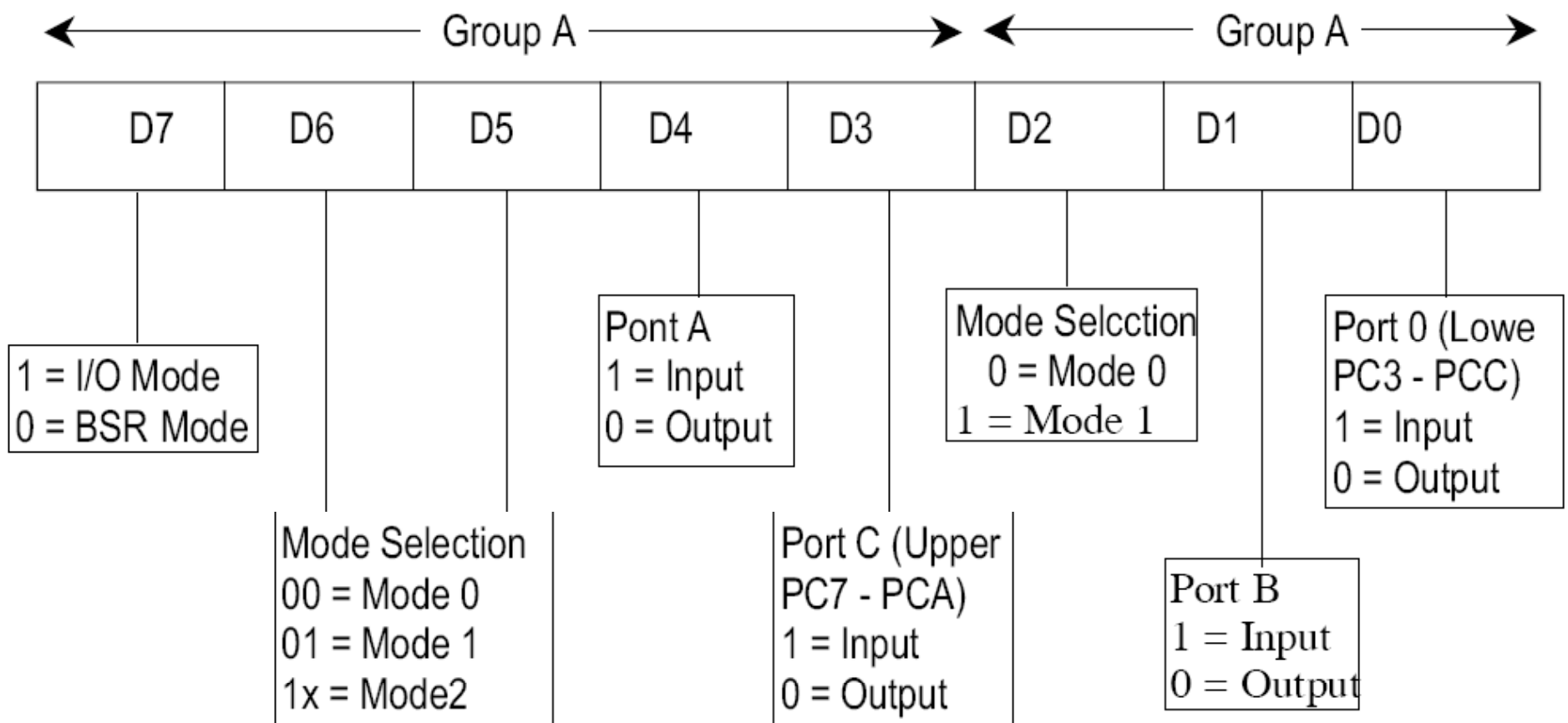
Các chế độ hoạt động IC 8255

/CS	A1	A0	Cổng
0	0	0	Port A
0	0	1	Port B
0	1	0	Port C
0	1	1	Thanh ghi điều khiển
1	X	X	8255 không được chọn

MỞ RỘNG CỔNG SONG SONG

Các chế độ hoạt động IC 8255

Thiết lập I/O Mode D7 = 1



MỞ RỘNG CỔNG SONG SONG

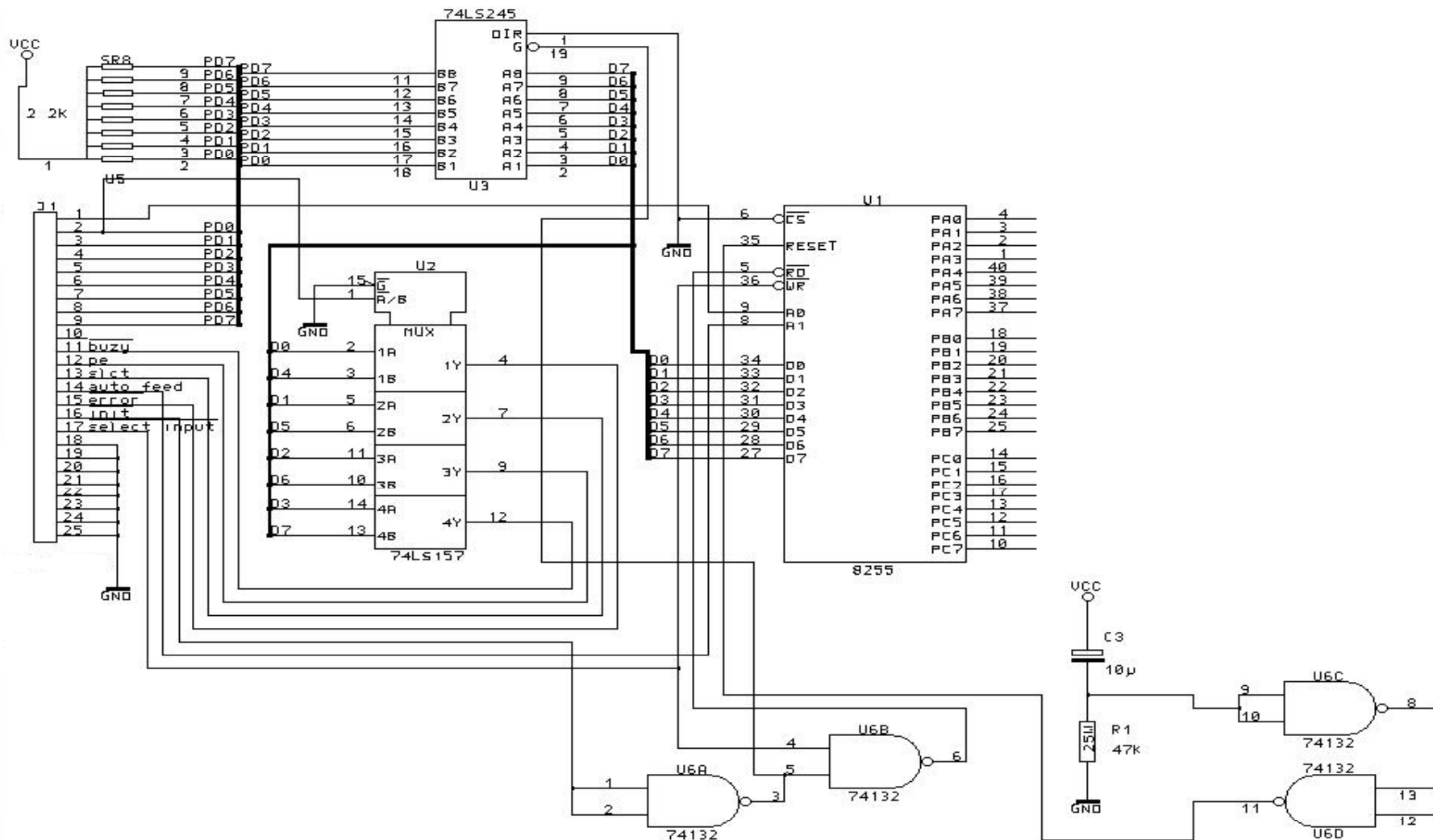
Các chế độ hoạt động IC 8255

Thiết lập BRS Mode D7 = 0



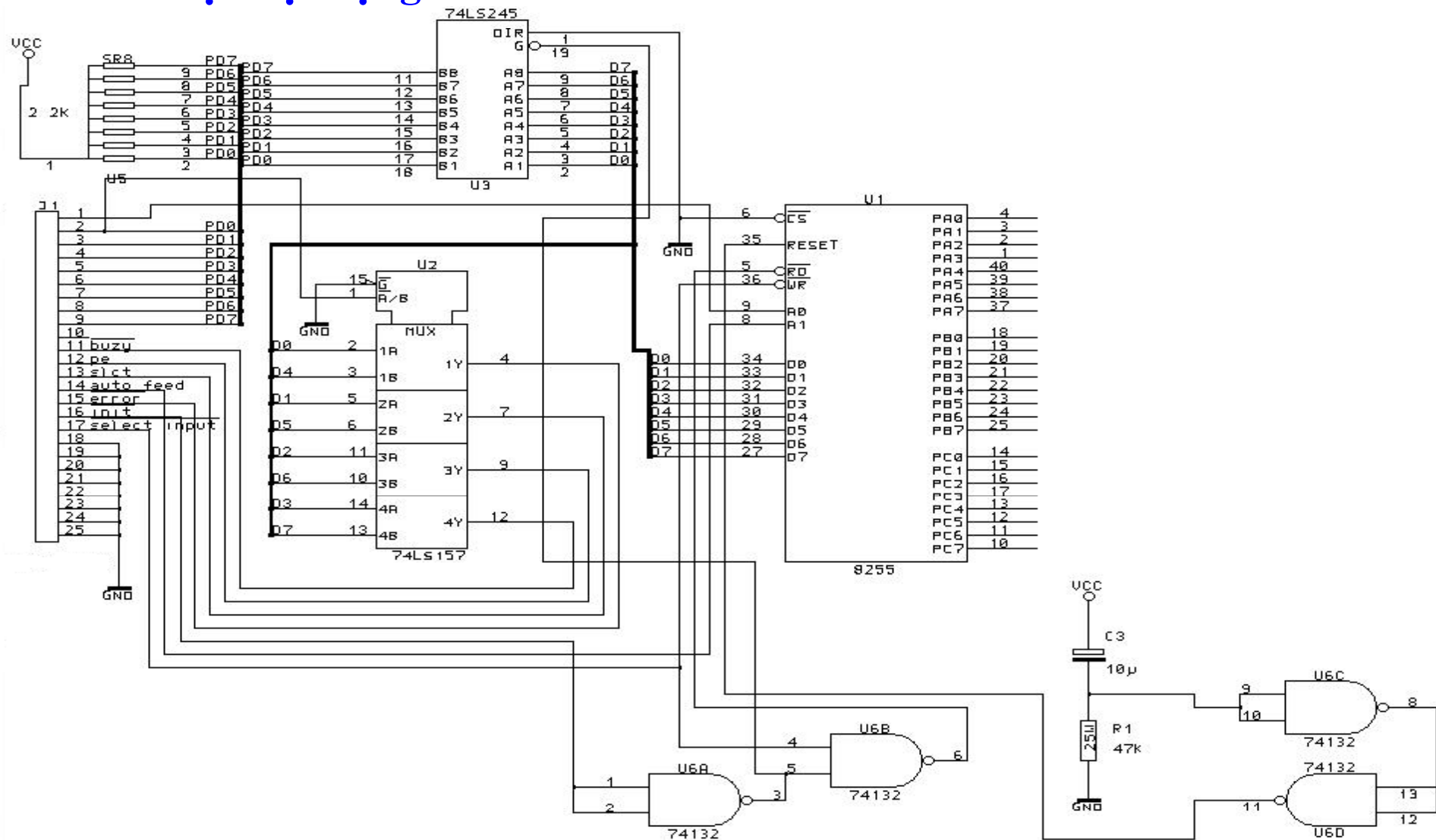
MỞ RỘNG CỔNG SONG SONG

Các chế độ hoạt động IC 8255



MỞ RỘNG CỔNG SONG SONG

Các chế độ hoạt động IC 8255



Tạo file .dll

VB thiếu một số hàm truy cập trực tiếp phần cứng và đa luồng. Để thực hiện những thiếu sót này, cần các file liên kết động (.dll) . Các file .dll này được viết từ những soft có hỗ trợ những truy cập này như VC, Delphi,...

Cách viết dll trên VC++

Các công cụ

Visual C++ 6 , Visual basic 6

Writing The DLL

Start VC++ IDE ,

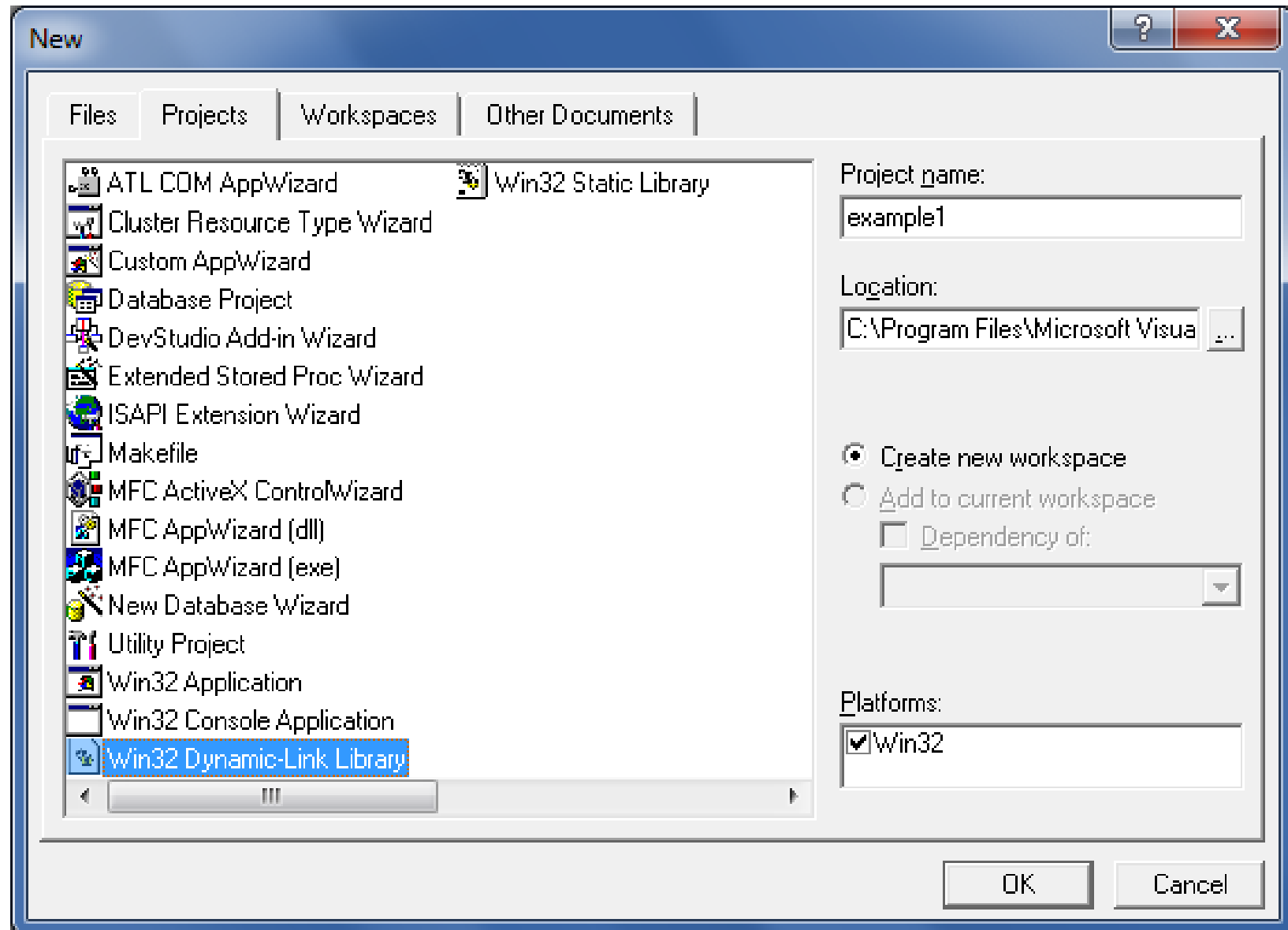
File menu → New.

Chọn Win32 Dynamic - link library from Projects tab.

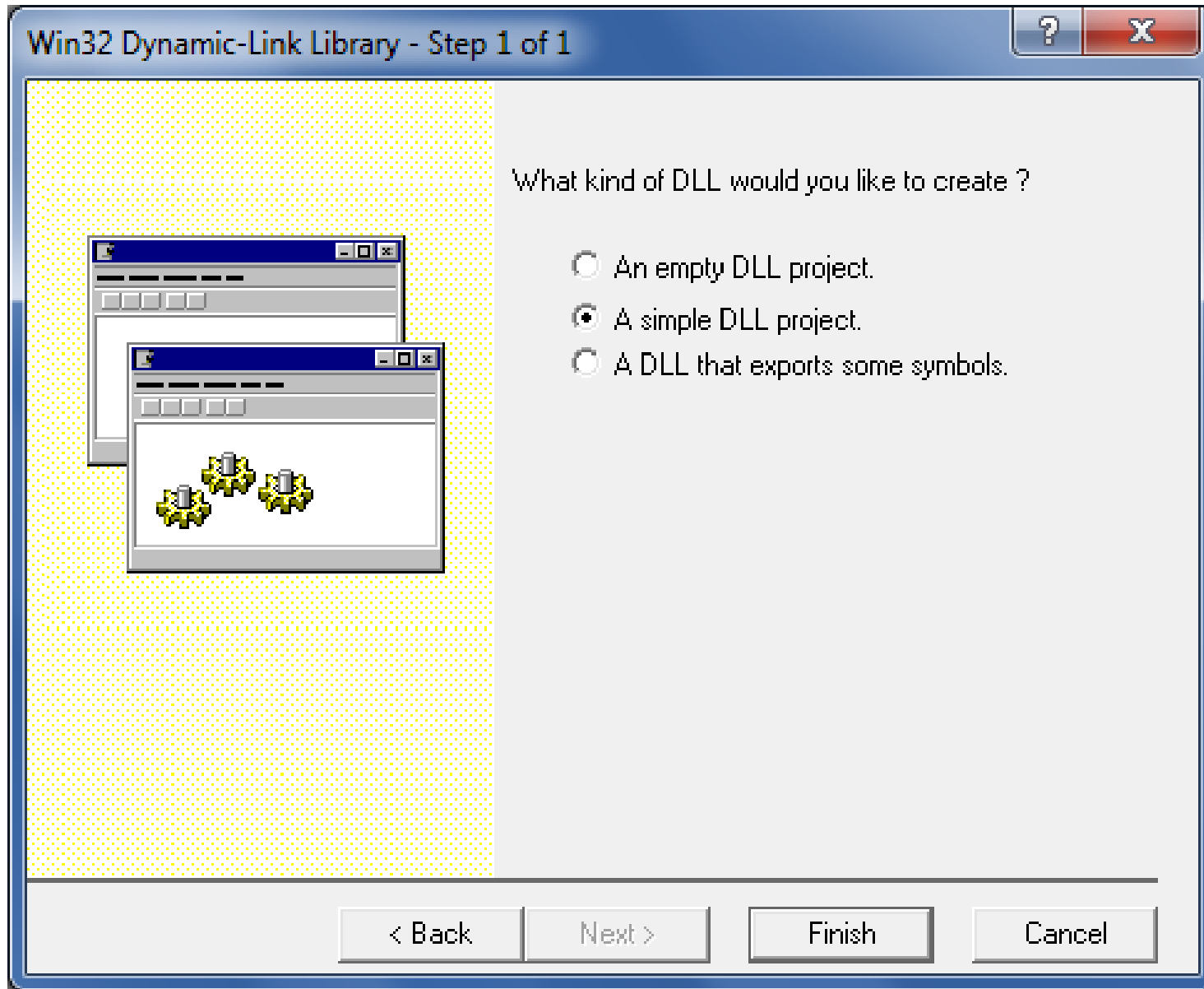
Đặt tên project name → click OK.

Chọn simple DLL project → click Finish

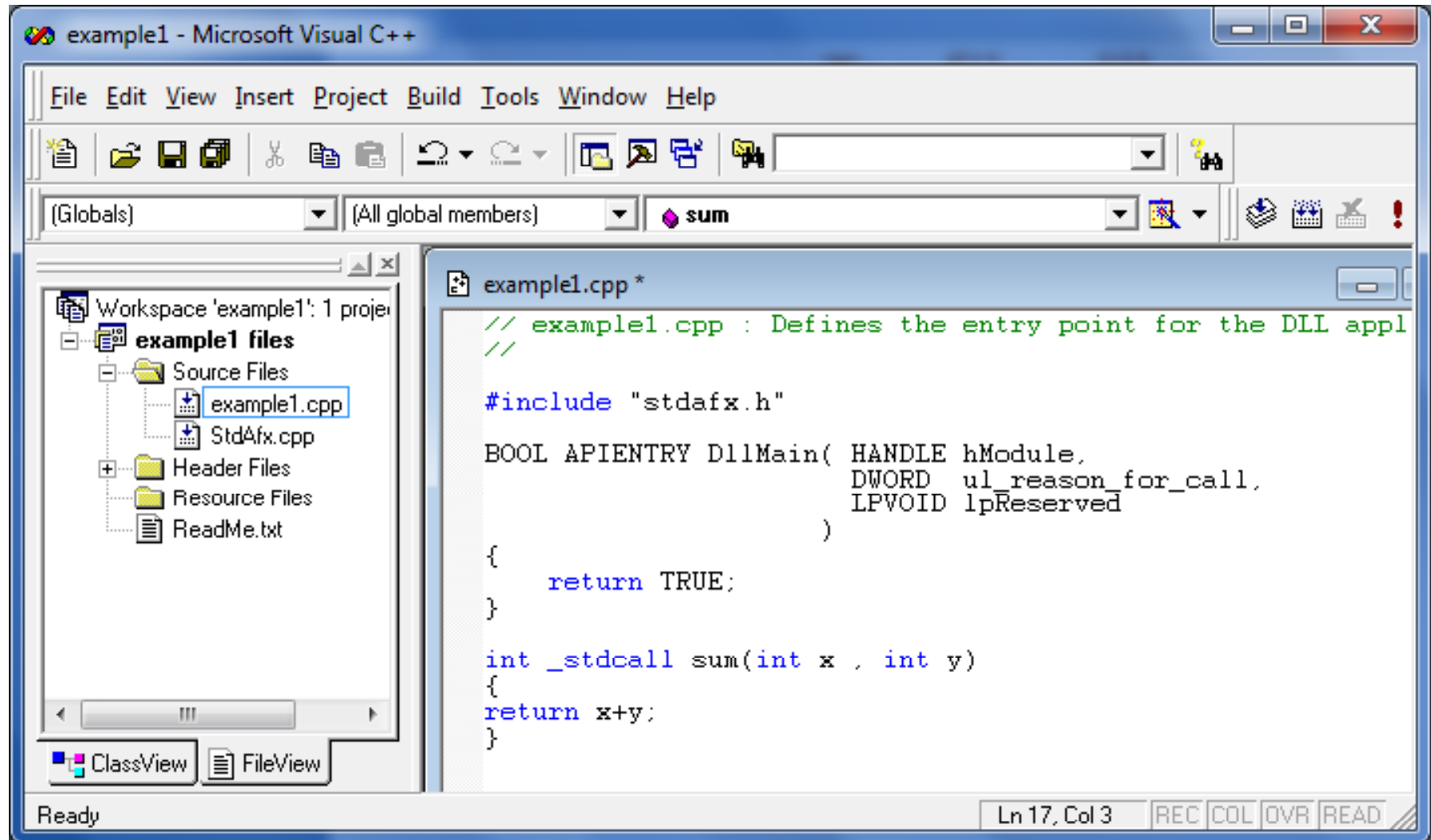
Tạo file .dll



Tạo file .dll



Tạo file .dll

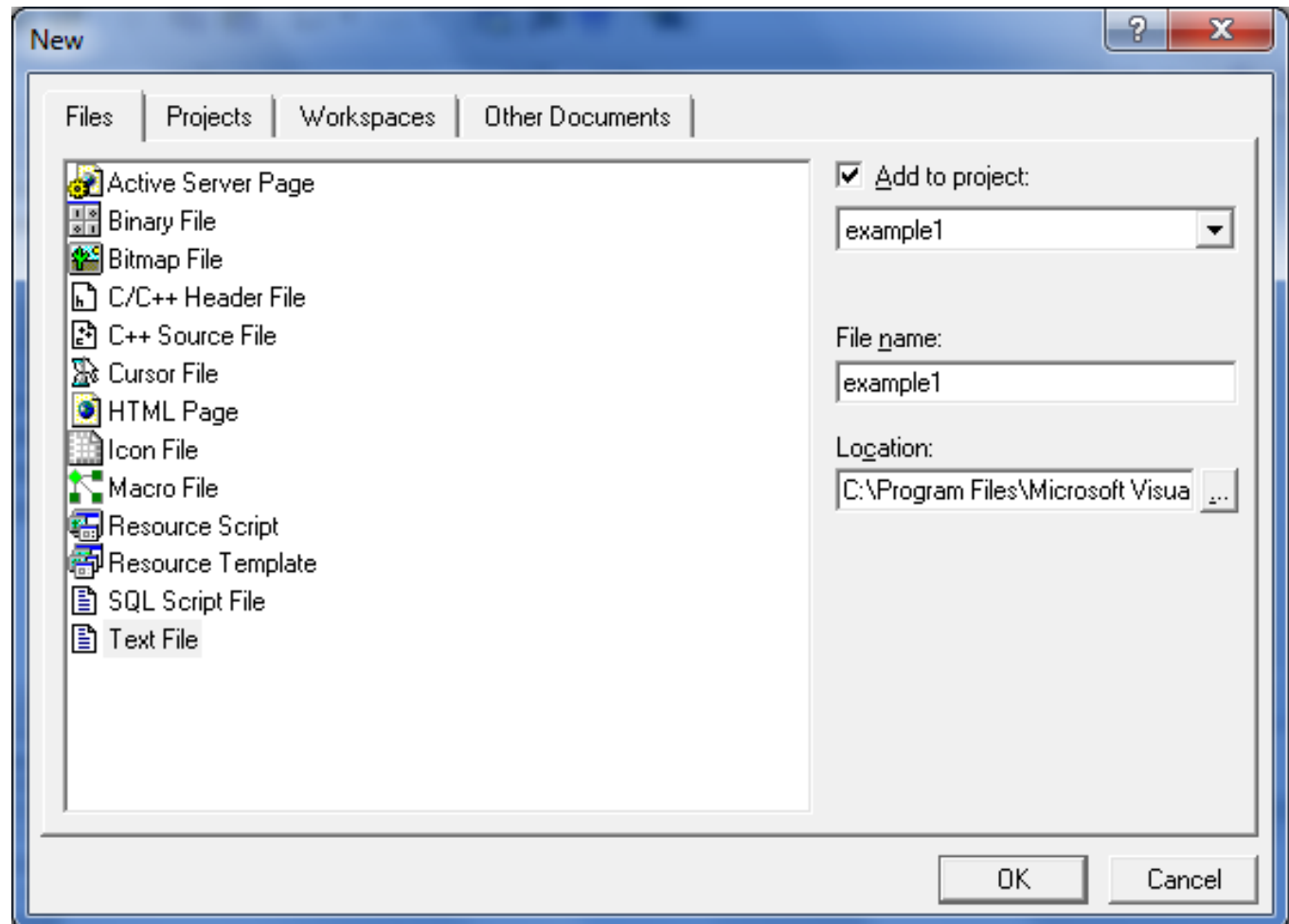


Tạo file .dll

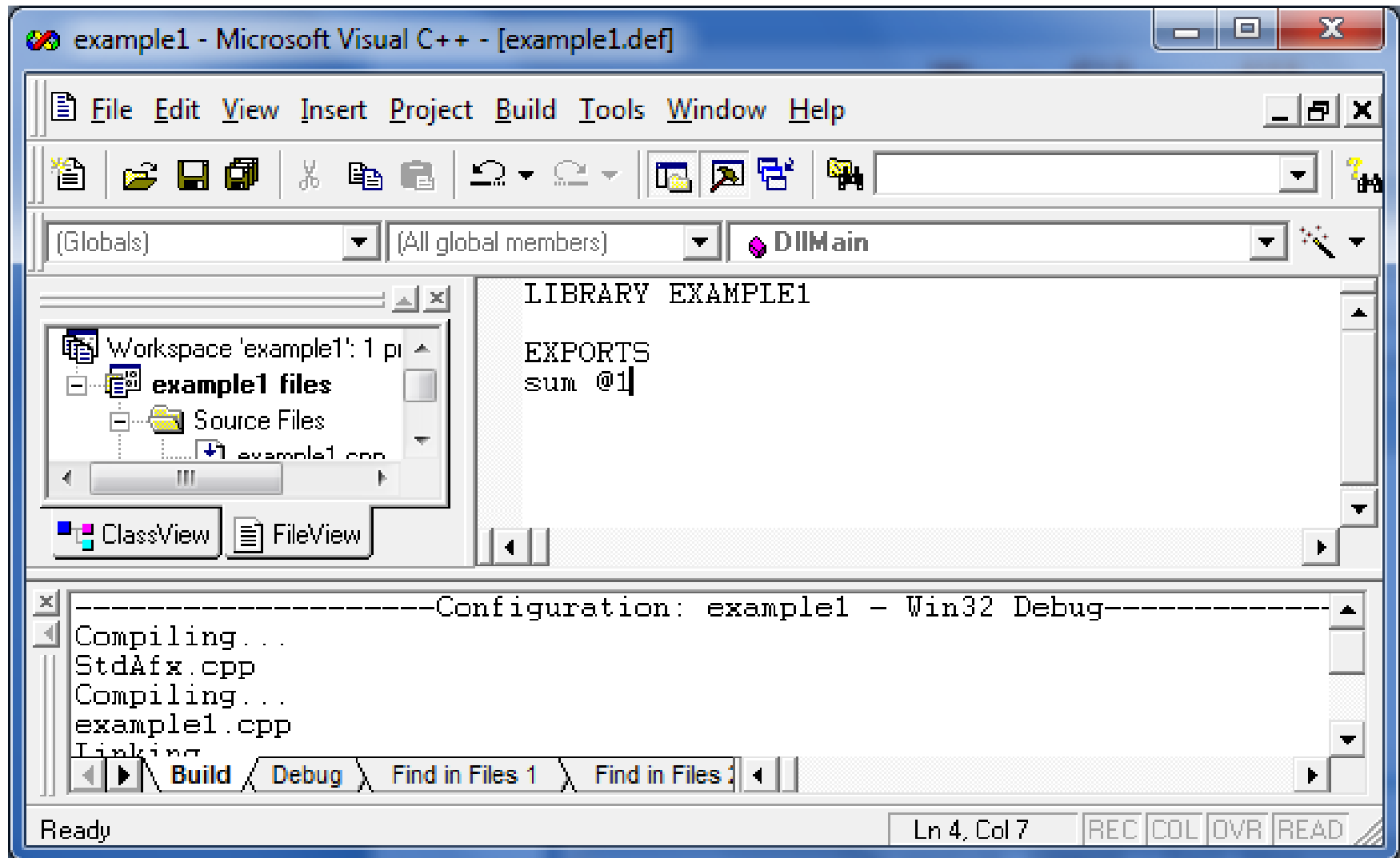
Now our function is ready. The next step is to create a '.def' file which informs the linker to export our function. Select New from File menu and then select text File from Files tab. Give it a name example1.def. Now you can see an empty file . Add these lines in to example1.def

Bước kế tiếp là tạo file .def cho phép xuất chương trình

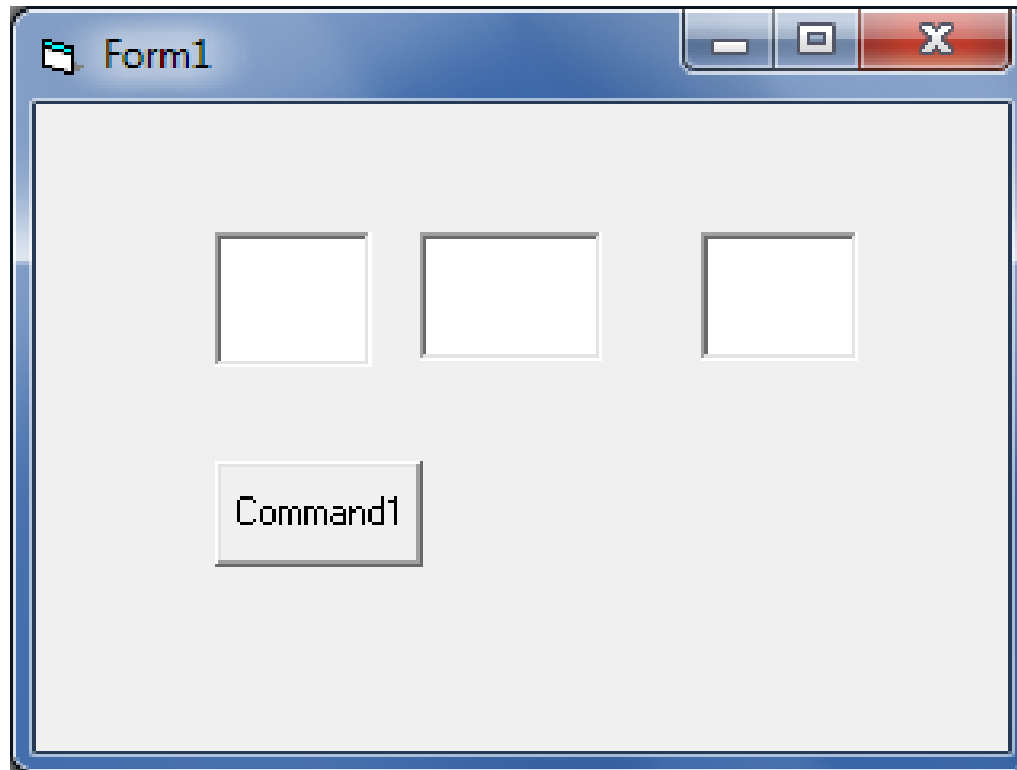
File → New → text File



Tạo file .dll



Tạo file .dll



Tạo file .dll

Private Declare Function sum Lib "example1.dll" (ByVal x As Long, ByVal y As Long) As Long

Private Sub Form_Load()

End Sub

Private Sub Command1_Click()

Text3.Text = Str(sum(CInt(Text1.Text), CInt(Text2.Text)))

End Sub

Tạo file .dll

