الفصل الحادى عشر 11

المخارج المتوازية

Parallel Ports

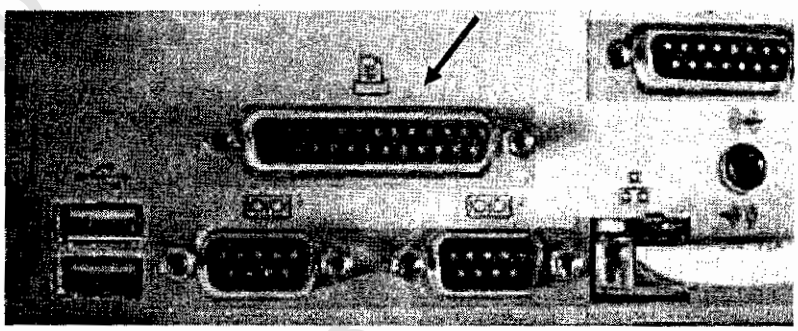
obeikandi.com

11-1 مقدمة

فى العادة يكون الحاسب مجهزا بمخرج متوازى واحد على الأقل ، ويمكن تجهيزه حتى 4 مخارج . فى العادة أيضا تأخذ هذه المخارج الأسماء المختصرة التالية : LPT1 و LPT2 و LPT3 و LPT4 ، وفى العادة يطلق على المخرج الأول الاسم PRN كبديل للاسم الأول وهذه خاصية للمخرج الأول فقط . الاسم LPT هو اختصار للعبارة Line Printer وذلك لأن هذه المخارج تستخدم فى العادة للتوصيل مع الطابعات . يمكن التعامل مع كل هذه المخارج بالأسماء السابقة من خلال نظام التشغيل DOS أو برنامج الإعداد Set Up . سنقدم فى هذا الفصل دراسة تفصيلية لهذه المخارج تمهيدا لاستخدامها كمخارج ومداخل للبيانات الرقمية من خلال لغات البرمجة مثل لغة التجميع assembly أو لغة C وغيرها وذلك كبديل سهل لعملية التعامل مع الحاسب من خلال المسارات القياسية التى سبق شرحها فى فصول سابقة .

نتيجة سرعة النقل العالية للبيانات على هذه المخارج فقد استخدمت فى الكثير من التطبيقات والتى منها التوصيل مع الراسمات Plotters ، والماسحات الضوئية Scanners ، وكذلك التوصيل بين جهازى حاسب لتبادل البيانات بينهما بسرعة كبيرة ، وأخيرا استخدامها فى إدخال البيانات وإخراجها من الحاسب فى أغراض التحكم المختلفة التى يحتاجها أى مستخدم ، وهذا ما سنركز عليه فى هذا الفصل حيث أننا لن نركز على مشاكل المصافحة مع الطابعة والتى ظهر بسببها أكثر من إصدار للمخرج المتوازى كان أولها المخرج المتوازى القياسى Standard Parallel Port, SPP ، الذى ظهر عام 1987 والذى تميز عن سابقه بإمكانية إدخال البيانات وإخراجها من الحاسب من خلاله . ثم المخرج المتوازى المطور Enhanced Parallel Port, EPP ، الذى ظهر عام 1991 والذى تميز عن سابقه بالسرعة العالية فى نقل البيانات التى تصل إلى 2 ميجابايت فى الثانية . هذه السرعات تكون هناك حاجة لها بالذات عند التعامل مع أوساط التخزين التى توصل من خلال هذا المخرج مثل الاسطوانات المدمجة والماسحات الضوئية وغيرها . ثم المخرج المتوازى ذو الإمكانيات الممتدة Extended Capability Parallel Port, ECP ، الذى ظهر عام 1992 لتحسين أداء الطابعات بالذات وليس الأجهزة الأخرى . كل هذه عبارة عن إصدارات للمخرج المتوازى تحقق سرعة أكثر فى نقل البيانات من خلال المصافحة المطورة . يحتوى هذا المخرج على 8 خطوط بيانات خرج ويمكن برمجتها كخطوط دخل ، وخمسة خطوط دخل ، و 4 خطوط ثنائية الاتجاه يمكن استخدامها للإدخال أو الإخراج . يمكن برمجة المخرج على أى جهاز حاسب ليعمل بأى واحدة من المواصفات السابقة من خلال ال BIOS إذا احتاج الأمر لذلك .

يوجد المخرج المتوازى فى ثلاث أنواع وكلها تأخذ الشكل D . النوع الأول ويسمى النوع A أو type A وله 25 طرف ، والنوع الثانى يسمى النوع B أو type B وله 36 طرف ، والنوع الثالث ويسمى النوع C أو type C وله 38 طرف وهو نفسه النوع الثانى بالضبط سوى إضافة طرفين يستخدمان لمعرفة إذا كانت الطابعة موصلة على المخرج أم لا . النوع الأول والثانى هما الأكثر شيوعا بالذات فى الحاسبات الشخصية هذه الأيام . هذا المخرج يعرف قياسيا بالمخرج IEEE1284 حيث توجد المواصفات القياسية (الخواص الكهربية والميكانيكية والأبعاد الهندسية) له تحت هذا المسمى . شكل (11-1) يبين منظرا عاما للمخرج المتوازى مع بعض المخارج الأخرى .



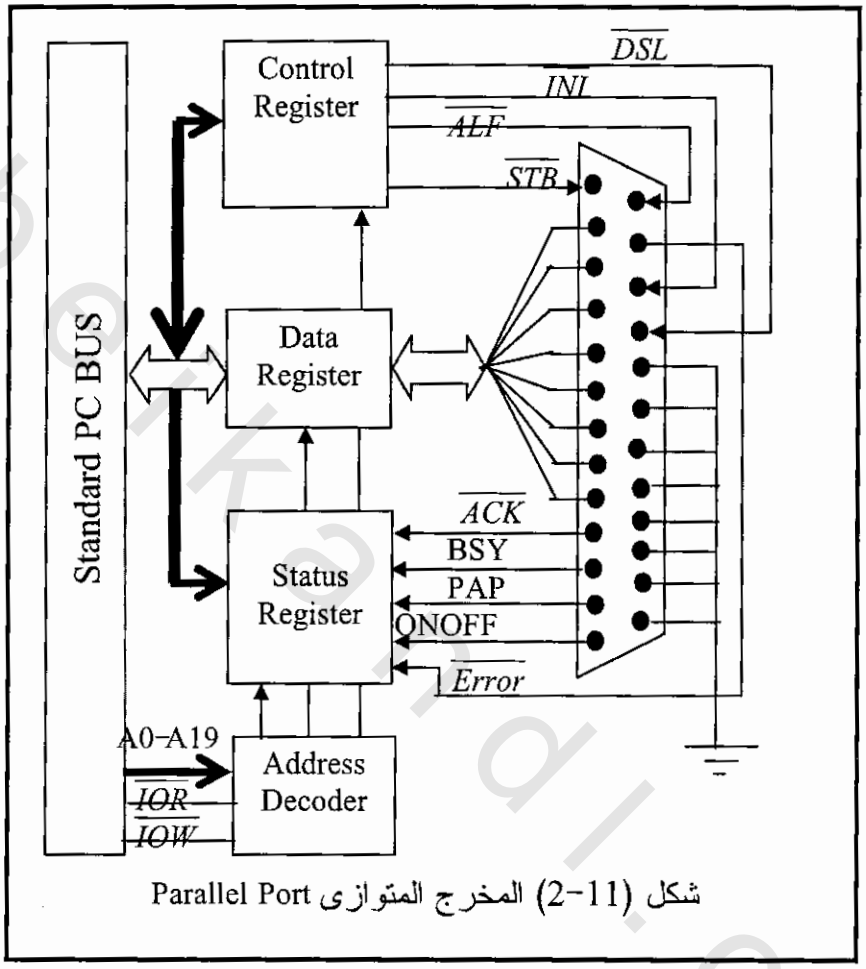
شكل (11-1) صورة المخرج المتوازى مع بعض المخارج الأخرى

11-2 تركيب المخرج المتوازى

شكل (11-2) يبين رسما تخطيطيا لمحتويات هذا المخرج ، وجدول 11-1 يبين نبذة مختصرة عن كل واحد من هذه الأطراف .

وجود شرطة أفقية فوق اسم أى إشارة تعنى أن هذا الإشارة تكون منخفضة الفعالية ، بمعنى أنها تكون فعالة عندما تكون صفر . وجود كلمة نعم فى عمود العكس فى جدول 11-1 تعنى أن هذه الإشارة تكون على طرف المخرج عكس ما هى عليه فى داخل المسجل . فمثلا إذا كان الطرف $\overline{STB}=0$ فإن ذلك يعنى أن البت المقابلة له فى المسجل الخاص بهذا الطرف وهو مسجل التحكم كما سنرى تساوى واحد . جدول 11-2 يبين عنوان كل مخرج من هذه المخارج ، وهذه العناوين قياسية فى كل الأجهزة ونلاحظ من هذا الجدول أن مسجل البيانات يأخذ دائما عنوان الأساس base address للمخرج . يتركب أى مخرج متوازى

من ثلاثة مسجلات أساسية يتم التعامل مع هذا المخرج من خلالها ، هذه المخارج سيتم شرحها ووظيفة كل منها فى الأجزاء التالية :



شكل (11-2) المخرج المتوازى Parallel Port

11-2-1 مسجل البيانات Data Register

أى بيانات يراد إخراجها على المخرج المتوازى توضع أولا فى مسجل البيانات. عنوان هذا المسجل هو عنوان المخرج ، أو أحيانا يطلق عليه عنوان الأساس للمخرج base address كما ذكرنا . نلاحظ من شكل (11-2) أن البيانات يتم إخراجها على الأطراف 2 حتى 9 ، وهذه الأطراف هى نقاط الاتصال بمسجل البيانات . الأمر التالى يخرج البايت 0xff على المخرج المتوازى الأول LPT1 والذى عنوانه 0x378 :

جدول 11-1

النوع A 25 طرف	النوع B 36 طرف	اسم الإشارة	اتجاه الإشارة	المسجل الذى تتبعه الإشارة	العكس
1	1	$\overline{Strobe}$	IN/OUT	Control	نعم
2	2	Data 0	IN/OUT	Data	
3	3	Data 1	IN/OUT	Data	
4	4	Data 2	IN/OUT	Data	
5	5	Data 3	IN/OUT	Data	
6	6	Data 4	IN/OUT	Data	
7	7	Data 5	IN/OUT	Data	
8	8	Data 6	IN/OUT	Data	
9	9	Data 7	IN/OUT	Data	
10	10	$\overline{ACK}$	IN	Status	
11	11	Busy	IN	Status	نعم
12	12	Paper	IN	Status	
13	13	ON OFF	IN	Status	
14	14	$\overline{ALF}$	IN/OUT	Control	نعم
15	32	$\overline{Error}$	IN	Status	
16	31	$\overline{INI}$	IN/OUT	Control	
17	36	$\overline{DSL}$	IN/OUT	Control	نعم
18-25	19-30	GND	GND		

```
outportb(0x378, 0xff);
```

حيث توضع البايت 0xff فى مسجل البيانات ومنه إلى الأطراف 2 حتى 9 الخاصة بالبيانات على هذا المخرج .

جدول 11-2

	مسجل البيانات DR	مسجل الحالة SR	مسجل التحكم CR
LPT1	0x378	0x379	0x37A
LPT2	0x278	0x279	0x27A
LPT3	0x3BC	0x3BD	0x3BE
LPT4	0x2BC	0x2BD	0x2BE

11-2-2 مسجل الحالة Status Register

هذا المسجل يعتبر مسجل قراءة فقط ، أى لا يمكن الكتابة فيه حيث يتم استقبال حالة الطابعة على هذه الخطوط . هذا المسجل يستخدم فقط خمس بتات منه وباقى البتات غير مستخدمه وهذه البتات متصلة بأطراف المخرج أرقام 10 و 11 و 12 و 13 و 15 كما فى شكل (11-2) . يمكن توضيح وظيفة كل بت من هذه البتات باختصار كما يلى :

البت $\overline{BSY}$

هذا الطرف يكون فعالا عندما يكون صفر ، ويعنى فى هذه الحالة أن الطابعة مشغولة فى طباعة حرف معين ، أو أن المخزن المؤقت لها buffer ممتلئ ، أو أن الطابعة فى حالة إعداد أو تجهيز initialization . فى هذه الحالة لا يمكن إرسال أى بيانات للطابعة حتى يصبح هذا الخط واحد . لاحظ أن هذه البت يظهر عكسها على الطرف رقم 11 للمخرج . أى أن الطرف 11 للمخرج سيكون واحد عندما تكون الطابعة مشغولة .

البت $\overline{ACK}$

بعد أن تستقبل الطابعة أى حرف فإنها ترسل نبضة اعتراف acknowledge بإتمام وصول هذا الحرف عن طريق وضع هذا الطرف (وبالتالى هذه البت) يساوى صفر ثم إرجاعه للواحد مرة أخرى ، زمن هذه النبضة هو 5 ميكروثانية تقريبا . هذه البت متصله بالطرف رقم 10 على المخرج .

البت PAP

عندما تكون هذا البت فعاله (1) فإن ذلك يعنى أن الطابعة تحتاج لأوراق ، أى أن الطابعة فرغت من الورق . فى هذه الحالة يتم تنشيط الطرف ONOF بجعله يساوى 1 للدلالة على أن الطابعة مفصولة ، وكذلك فإن الطرف Error يتم تنشيطه بجعله صفر للدلالة على أن هناك خطأ معين .

الطرف ONOF

ONOF=1 ⟵ الطابعة موصلة Online

ONOF=0 ⟵ الطابعة غير موصلة Offline

البت $\overline{Error}$

Error=0 ⟵ هناك خطأ

Error=1 ⟵ ليس هناك خطأ

شكل (11-3) يبين بتات مسجل الحالة والأطراف المتصلة به على أطراف المخرج .

7	6	5	4	3			
$\overline{BSY}$	$\overline{ACK}$	PAP	ONOF	$\overline{Error}$	X	X	X
11	10	12	13	15	--	--	--

رقم الطرف على المخرج

شكل (11 -3) مسجل الحالة للمخرج المتوازى

11-2-3 مسجل التحكم Control Register

مسجل التحكم ثنائى الاتجاه ، أى يمكن القراءة منه والكتابة فيه . يمكن التحكم فى أداء الطابعة من خلال البيانات التى ترسل لهذا المسجل . وظيفة كل بت من بتات هذا المسجل يمكن شرحها كما يلى :

البت IRQ

عند تنشيط هذا الطرف بجعله يساوى واحد يمكن مقاطعة المعالج من خلال قناة المقاطعة IRQ7 فى حالة المخرج LPT1 و IRQ5 فى حالة المخرج LPT2 .

IRQ=1 ←— المقاطعة نشطة

IRQ=0 ←— المقاطعة غير نشطة

البت DSL

بت اختيار الطابعة Device Select Line, DSL :

DSL=1 ←— الطابعة تم اختيارها selected

DSL=0 ←— الطابعة غير مختارة

البت $\overline{INI}$

عندما تكون هذه البت فعاله (0) فإن الطابعة تكون فى حالة إعداد أو تهيئة Initialization ويتم تفريغ المخزن المؤقت للطابعة ، ويجب أن يظل هذا الطرف صفر لمدة 50 ميكروثانية على الأقل :

$\overline{INI}$ =1 ←— الطابعة تعمل فى وضعها الطبيعى

$\overline{INI}$ =0 ←— الطابعة فى حالة استعداد أو تهيئة

البت ALF

هذه البت فعاله (1) عندما تكون الطابعة فى حالة تغذية سطر جديد Automatic Line Feed :

ALF=1 ⟵ الطابعة فى حالة تغذية سطر جديد
ALF=0 ⟵ الطابعة ليست فى حالة تغذية سطر جديد

البت STB

عند إرسال حرف للطابعة فإن هذا الحرف يوضع أولا فى مسجل البيانات ثم نرسل نبضة تسجيل (واحد ثم صفر ثم واحد) إلى الطابعة للدلالة على أن هناك حرف مطلوب طباعته . لذلك فإن النبضة STB تخبر الطابعة بوجود حرف جاهز للطباعة . لاحظ أن البت STB عكس الطرف $\overline{STB}$.

STB=1 ⟵ هناك بيانات تم إرسالها للطابعة
STB=0 ⟵ ليس هناك بيانات مرسلة للطابعة

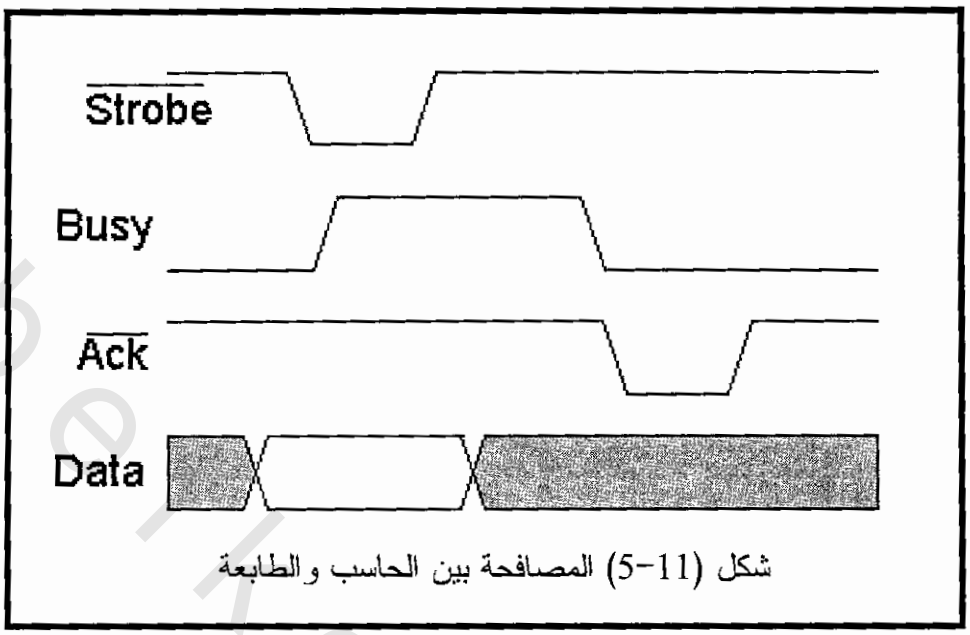
شكل (11-4) يبين تفاصيل مسجل التحكم والأطراف الخاصة بها على المخرج المتوازى .

7	6	5	4	3	2	1	
X	X	X	IRQ	DSL	INI	ALF	S
--	--	--	--	16	18	14	1

رقم الطرف على المخرج

شكل (11-4) مسجل التحكم للمخرج المتوازى

شكل (11-5) يوضح عملية المصافحة التى تتم بين الطابعة والحاسب . يقوم الحاسب أولا بمراقبة الخط BUSY فإن كان هذا الخط صفر فإن ذلك يعنى أن الطابعة على استعداد لاستقبال حرف جديد لأنها غير مشغولة . فى هذه الحالة يقوم الحاسب بوضع البيانات على الخطوط D0 إلى D7 وإحداث نبضة (واحد صفر واحد) لمدة واحد ميكروثانية تقريبا على الخط $\overline{STB}$ لإخبار الطابعة بوجود بيانات على خطوط البيانات . عندما تشعر الطابعة بالنبضة على الخط Strobe تقوم بوضع الخط Busy=1 لتخبر الحاسب أنها مشغولة فلا يرسل أى بيانات أخرى ، إلى أن تنتهى من طباعة الحرف فتضع الطرف Busy=0 مرة أخرى ، وتحدث نبضة (واحد ثم صفر ثم واحد) على الخط $\overline{ACK}$ لإخبار الحاسب أنها انتهت من طباعة الحرف وعلى استعداد لتلقى حرف جديد . لإسراع عملية نقل البيانات يتم الاستغناء عن إشارة الاعتراف فى عملية المصافحة ، بل إن المخارج ذات الإمكانيات الممتدة Extended Capability Ports, ECP تترك عملية المصافحة بالكامل للدوائر الخاصة بالمخرج وليس من خلال البرمجة كما سبق ، وكل ذلك لإسراع عملية نقل البيانات بين الحاسب والطابعة .



شكل (11-5) المصافحة بين الحاسب والطابعة

البرنامج التالى يرسل حرفا للطابعة :

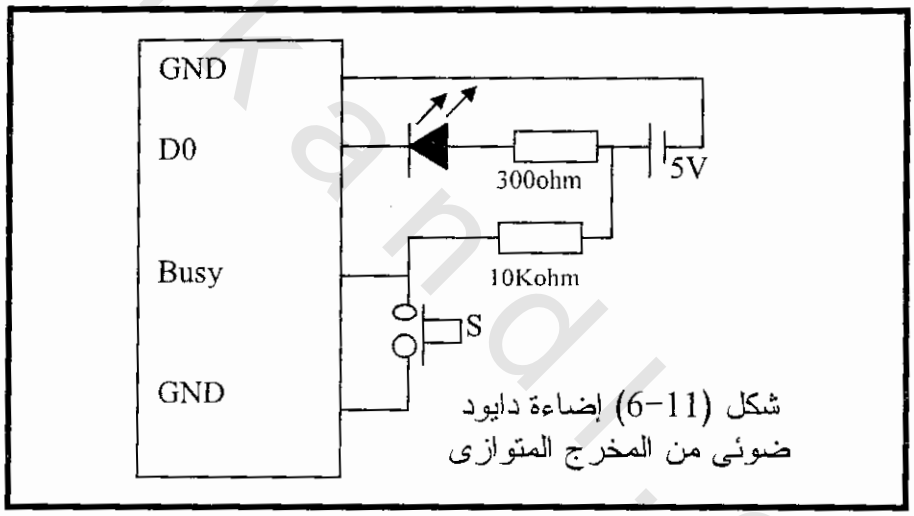
```
main()
{ int i, code;
  do {
      code=inportb(0x379); قراءة مسجل الحالة
       code=code&0x80;  BSY اختبار الطرف
     } while (code !=0x80);
  outportb(0x378, (int) character); إرسال الحرف المراد طباعته
  code=inportb(0x37a);  قراءة مسجل التحكم
  code=code | 0x01;  STRB  تنشيط الطرف
  outportb(0x37a,code);
  for(i=0; i<wait; i++); STR  انتظار لبعض الوقت أثناء تنشيط الطرف
  code=inportb(0x37a);
  code=code & 0xfe; STR  إخماد الطرف
  outport(0x37a,code);
  do {
      code=inportb(0x379); قراءة مسجل الحالة
      code=code & 0x40; ACK  اختبار الطرف
      }while(code!=0x00);
                          بذلك تكون الطابعة جاهزة لاستقبال حرف جديد ;
}
```


obeikandl.com

11-3 إخراج البيانات من الحاسب خلال المخرج المتوازى

شكل (11-6) يبين إضاءة دايود ضوئى LED بطريقة متقطعة حيث يكون الدايود مضيئا لمدة ثانية ويطفىء لمدة ثانية طالما أن الزرار S مضغوط ، عندما يكون الزرار غير مضغوط فإن الدايود يطفىء .

فى شكل (11-6) لكى نضىء الدايود نضع صفر على خط البيانات D0 ، ولكى نطفؤه نضع واحد على هذا الخط . عند ضغط الزرار S فإن الطرف Busy يصبح صفر ونقرأه واحد فى البت المقابلة لمسجل الحالة لأن هذا الطرف معكوس كما ذكرنا . عندما يكون الزرار فاتح (غير مضغوط) فإن الطرف Busy سيكون واحد ونقرأه صفر داخل مسجل التحكم . لذلك فإن البرنامج التالى سيقرأ مسجل الحالة ويختبره ليرى إذا كان الطرف Busy واحد أم صفر وعلى ضوء ذلك سيضىء الدايود أم لا . لكى نخرج من البرنامج نضرب أى مفتاح .



شكل (11-6) إضاءة دايود ضوئى من المخرج المتوازى

```
#include<stdio.h>
#define DATA 0x378
#define STATUS DATA+1
#define CONTROL DATA+2
main()
{ int n1
  do{ n1=inportb(STATUS);
      if (n1&0x80==1) /* لم تضغط بالفتيح*/
      { outportb(DATA, 0x00);
        delay(1000);
```

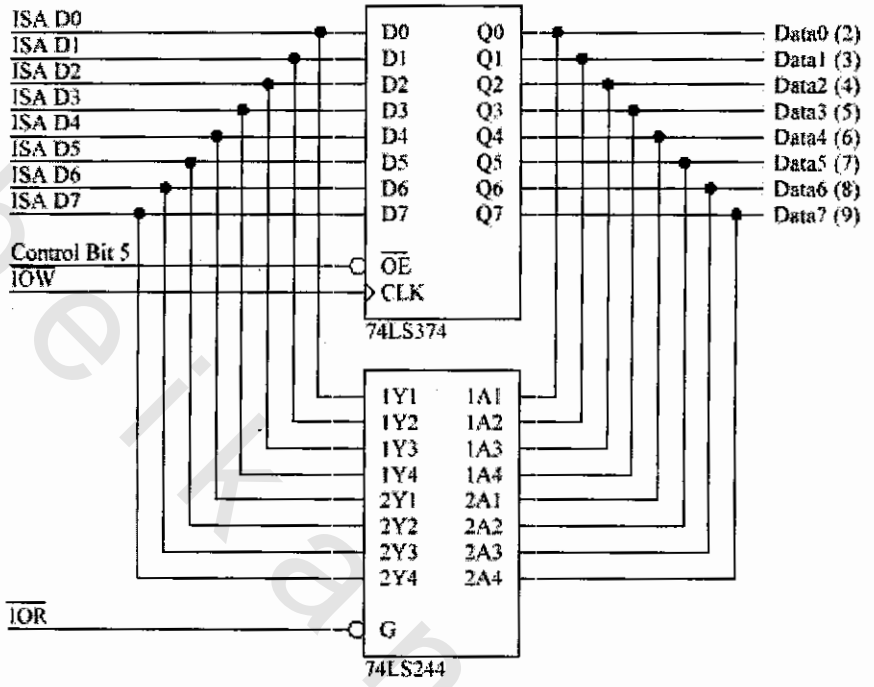
```
        outportb(DATA,0x01);
        delay(1000);
      }
      else
          outportb(DATA,0x01);
    }while (!kbhit());
}
```

11-4 إدخال البيانات إلى الحاسب من المخرج المتوازى

كما رأينا فإن عملية إخراج بيانات على خطوط البيانات D0 إلى D7 تعتبر عملية سهلة حيث بأمر إخراج واحد يمكن إخراج أى بيانات على خطوط المخرج . المشكلة هى فى توظيف هذا المخرج المتوازى لإدخال بيانات إلى الحاسب . شكل (11-7) يبين مكونات المخرج المتوازى القياسى Standard Parallel Port, SPP . فى هذا الشكل البت رقم 5 من مسجل التحكم موصلة على طرف تنشيط الخرج للشريحة 74373 وهو الطرف $\overline{OE}$. هذه البت تسمى بت تنشيط خاصية ثنائية الاتجاه Bidirectional property bit . عندما تكون هذه البت تساوى صفر فإن البيانات يمكنها أن تخرج من الحاسب إلى أطراف المخرج D0 إلى D7 . بينما إذا كانت هذه البت تساوى واحد فإن خرج الشريحة 47373 سيكون مقاومة عالية high impedance ، وفى هذه الحالة يمكن لأى بيانات مدخلة على الأطراف D0 إلى D7 أن تدخل إلى الحاسب بأى أمر قراءة الذى ينشط الخط $\overline{IOR}$ القادم من الحاسب والذى بدوره ينشط الشريحة 74244 . هذه الخاصية موجودة فى كل الحاسبات الحديثة هذه الأيام ويمكن من خلال هذه الخاصية إدخال وإخراج البيانات .

المشكلة هنا أنه ليست كل المخارج المتوازية توفر خاصية ثنائية الاتجاه السابقة. فى هذه الحالة يكون الطرف $\overline{OE}$ الخاص بالشريحة 74373 موصلا دائما على الأرضى بحيث أن آخر بيانات يتم إخراجها ستظل ممسوكة على خطوط البيانات D0 إلى D7 دائما . وفى هذه الحالة فإنه عند وضع بيانات خارجية على هذه الخطوط تمهيدا لإدخالها إلى الحاسب ستحدث مشاكل نتيجة تعارض الإشارتين الداخلة والخارجة على هذه الخطوط . أما إذا حاولنا قراءة خطوط البيانات D0 إلى D7 بأى أمر قراءة فى هذه الحالة (حالة عدم وجود خاصية ثنائية الاتجاه) فإننا سنقرأ آخر بيانات تم إخراجها على المخرج نتيجة دخول البيانات مرة أخرى من خلال الشريحة 74244 . ليس أمامنا سوى استخدام جهاز قياس متعدد

الوظائف مثل الأفوميتر multimeter أو المجس المنطقى logic probe لاختبار سلوك خرج المخرج مع الحالات المختلفة للبت الخامسة فى مسجل التحكم .



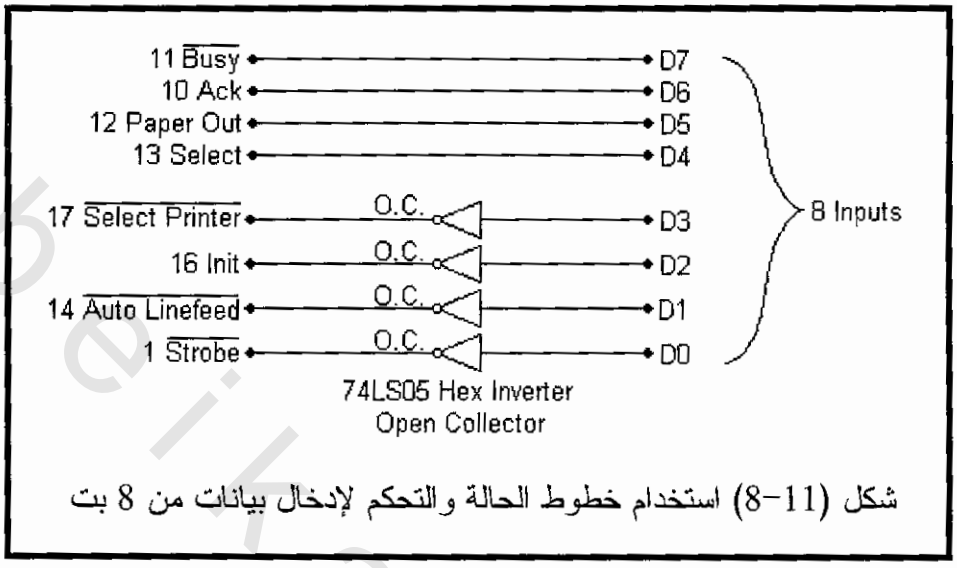
شكل (11-7) خاصية الخرج والدخل فى المخرج المتوازى القياسى SPP

المشكلة الأخرى أيضا هى أن بعض المصنعين يسلك طرق أخرى فى توفير خاصية ثنائية الاتجاه ، فالبعض مثلا ينشط اتجاه الخرج بالبت رقم 5 وينشط اتجاه الدخل بالبت رقم 6 ، وقد تجد طرقا أخرى . لذلك فإن أنسب وسيلة لمعرفة ذلك هى استخدام أجهزة القياس مع المحاولة والخطأ وذلك إذا لم يكن هناك كتالوج خاص بجهاز الحاسب موضحا به الموقف من هذه الخاصية .

الطريقة الأخرى لإدخال بيانات إلى الحاسب من خلال المخرج المتوازى هى من خلال خطوط الحالة (5 خطوط) وخطوط التحكم (4 خطوط) ، أى أنه لدينا 9 خطوط يمكن استخدامها لإدخال البيانات . شكل (11-8) يبين استخدام 8 من هذه الخطوط كخطوط إدخال .

فى هذا الشكل لابد من استخدام بوابات عكس ذات مجمع مفتوح open collector inverter على خطوط التحكم لأنها أصلا تخرج من مسجل التحكم على مجمع مفتوح . الخطوط التى عليها شرطة تكون معكوسة فى البت المقابلة فى المسجل.

من المفيد أن تجعل كل خطوط التحكم التى سيتم قراءتها تساوى واحد عن طريق إخراج البايت xxxx0100 إلى مسجل التحكم قبل القراءة باستخدم امر كالتالى :



شكل (11-8) استخدام خطوط الحالة والتحكم لإدخال بيانات من 8 بت

```
outportb(CONTROL,inportb(CONTROL) & 0xF0 | 0x04);
```

حيث control تعنى رقم مسجل التحكم . لاحظ أن التعبير inport(control)&0xF0 عندما يتم تنفيذه سيترك الأربع بتات ذات القيمة العظمى من مسجل التحكم كما هى دون تغيير وأما البايت ذات القيمة الصغرى فستصبح كلها أصفار . بعد ذلك عندما نعمل OR للناتج مع الرقم 04 فإن ذلك سيضمن أن الأربع بتات ذات القيمة الصغرى من مسجل التحكم ستكون وحايد . تذكر أن البتات رقم صفر وواحد وثلاثة فى الرقم 04 تكون معكوسة عند دخولها إلى المسجل .

الآن يمكن قراءة النصف الثانى من مسجل الحالة باستخدام الأمر التالى :

```
a = (inportb(STATUS) & 0xF0); /* Read MSnibble */
```

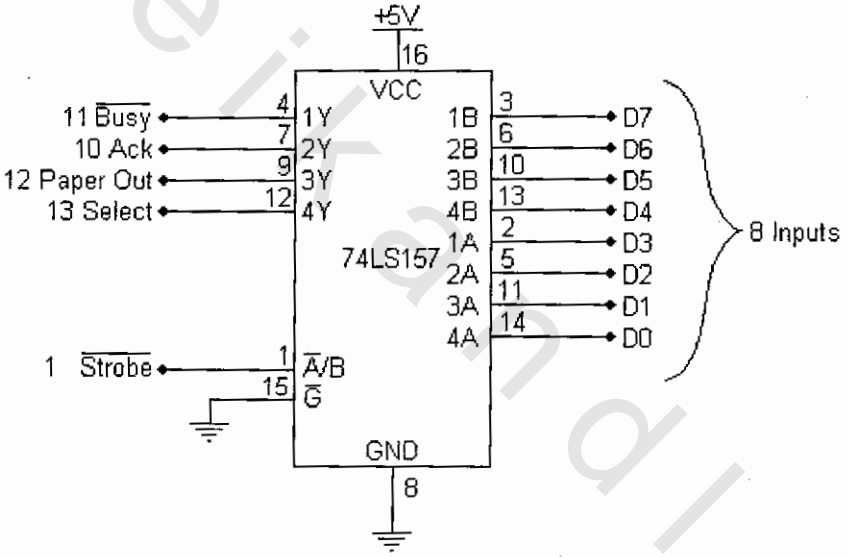
لاحظ العملية AND لمسجل الحالة مع الرقم F0 للتعامل مع النصف الثانى فقط من المسجل . الآن يمكن قراءة النصف الأول من مسجل التحكم باستخدام الأوامر التالية :

```
a = a|(inportb(CONTROL)&0x0F);/* Read LSnibble */
a = a ^ 0x84; /* Toggle Bit 2 & 7 */
```

لاحظ أيضا استخدام عملية AND مع الرقم 0F للتعامل مع النصف الأول من المسجل فقط . بعد ذلك تم استخدام عملية OR فى الأمر الأول أيضا حتى نركب نصفى مسجل الحالة الثانى مع نصف مسجل التحكم الأول . الأمر الثانى استخدم

عملية XOR للمتغير a وهو البيانات المدخلة مع الرقم 84H لعكس البت السابعة والثانية لأنهما سينعكسان عند الدخول على المسجل .
الطريقة الأخيرة لإدخال بيانات إلى الحاسب من خلال المخرج المتوازى والتى نعتقد أنها الأضمن وبها نتجنب الكثير من صعوبات التعامل مع مسجل التحكم إلا أنها تكون أبطأ . هذه الطريقة تتمثل فى استخدام 4 خطوط فقط من مسجل الحالة وقراءة البايت المدخلة على مرتين . شكل (11-9) يبين دائرة بسيطة مقترحة لذلك والبرنامج المستخدم لقراءة الدخل فى هذه الحالة سيكون كالتالى :

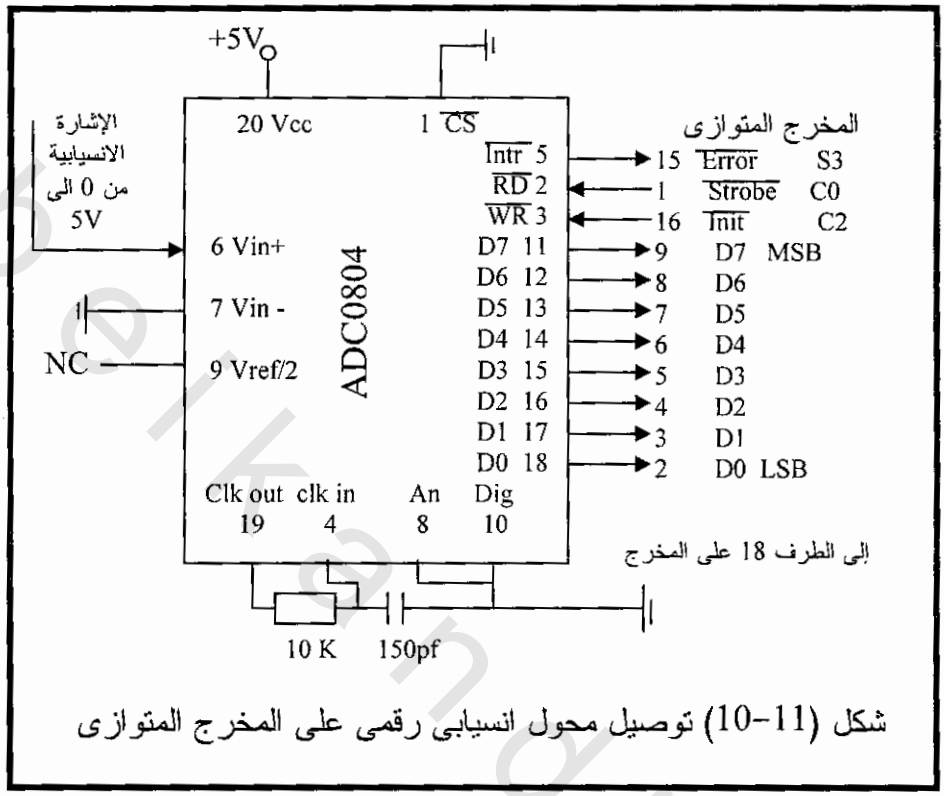
```
outportb(CONTROL, inportb(CONTROL) | 0x01);
```



شكل (11-9) إدخال بايت من البيانات على نصفين من خلال مسجل الحالة فى المخرج المتوازى

```
do{lc++}
        while(((inportb(status)&08)==0) && (lc!=256));
if (lc==256) cout<< "Time out, Hardware not connected or powered or
                        functioning ";
    else
    { outportb(control, 27); // تنشيط خرج المحول
      value=inportb(data);
     outportb(control, 26); // إخماد خرج المحول
      cout << "ADC value= " << (unsigned int) value << endl;
```

```
    }
    هنا تمت قراءة خرج المحول وطباعتها على الشاشة ويمكن الاستفادة بها //
}}
```



شكل (11-10) توصيل محول انسيابى رقمى على المخرج المتوازى

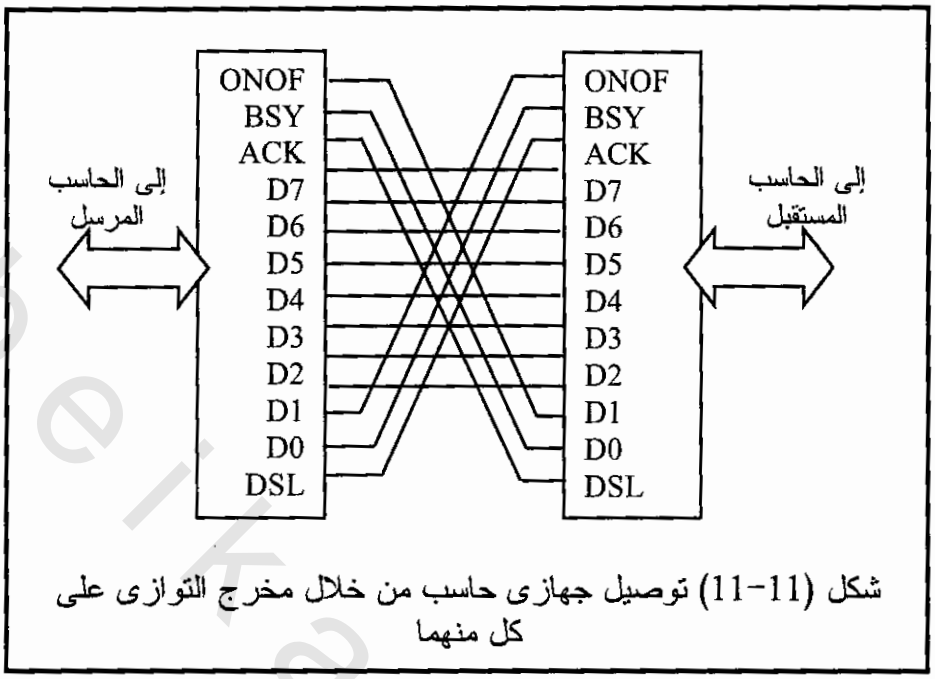
لاحظ أنه من الضرورى هنا استخدام حلقات تأخير بعد تنشيط الطرف WR وأيضا بعد تنشيط الطرف RD بالذات مع الحاسبات السريعة التى قد لا تتوافق سرعتها مع سرعة المحول البطيئة .

من التطبيقات الظريفة للمخارج المتوازية هى إمكانية استخدامها لتبادل البيانات بين جهازى حاسب . المخارج المتوالية تستخدم أساسا لهذا الغرض ولكنها تكون أبطأ من استخدام المخارج المتوازية لأننا هنا يمكننا نقل بايت كاملة فى كل مرة وليس بت واحدة كما فى حالة الاتصال المتوالى كما سنرى فيما بعد . شكل (11-11) يبين كيفية توصيل جهازى الحاسب من خلال المخرج المتوازى على كل منهما .

نلاحظ من شكل (11-11) أن أطراف البيانات فى كل من المخرجين تم توصيلها مع بعضها مباشرة . أما الأطراف STB (1) و DSL (17) و INI (16) فى الجهاز الأول فقد تم توصيلها بالأطراف ACK (10) و OFON (13) و

BSY (11) فى الجهاز الثانى كما هو مبين . وبنفس الطريقة العكسية تم توصيل الأطراف STB (1) و DSL (17) و INI (16) فى الجهاز الثانى بالأطراف ACK (10) و OFON (13) و BSY (11) فى الجهاز الأول ، لم يتم وضع خط فوق الخطوط المعكوسة . أما عملية الاتصال فتتم كما يلى :

- يقوم المرسل بوضع صفر على الطرف 17 للمخرج عن طريق وضع واحد فى البت 3 (DSL) من مسجل التحكم .
- عندما يقرأ المستقبل هذه الإشارة على الطرف 13 لمخرجه وهو البت 4 من مسجل الحالة يفهم أن المرسل يريد التعامل معه ، فيقوم بتنشيط الطرف INI بوضع واحد فى البت الثانية من مسجل التحكم .
- يشعر المرسل بتنشيط الطرف INI من خلال قراءة صفر على البت 4 من مسجل الحالة . وبذلك يكون قد تم التوصيل بين الجهازين .
- بذلك يستطيع المرسل أن يضع البايت المراد إرسالها إلى الطرف الآخر فى مسجل البيانات ، ثم يضع الخط STB يساوى صفر عن طريق وضع 1 فى البت رقم 0 من مسجل التحكم . بعد ذلك يضع 1 على الطرف STB مرة أخرى بوضع صفر فى البت 0 من مسجل التحكم . بذلك يتم وضع نبضة قصيرة 101 على الطرف STB . نتيجة هذه النبضة القصيرة على الطرف STB تبدأ عملية إرسال البيانات من المرسل وفى هذه اللحظة يعطى طرف المقاطعة نبضة مقاطعة على الطرف IRQ .
- عند وصول نبضة المقاطعة للمستقبل يبدأ برنامج خدمة المقاطعة فى التنفيذ حيث يقرأ البيانات الموجودة على أطراف البيانات ويخزنها فى الذاكرة أو فى مخزن مؤقت أو يستخدمها فى أى غرض يريد .
- بعد ذلك يقوم المرسل بوضع صفر ثم واحد على الطرف STB الخاص به ، وذلك عن طريق وضع واحد ثم صفر فى البت 0 فى مسجل التحكم . هذه النبضة القصيرة على الطرف STB تمثل نبضة اعتراف باستقبال البيانات عن طريق المرسل وتولد نبضة مقاطعة IRQ على طرف المقاطعة الخاص بالمستقبل .
- يمكن استخدام نبضة المقاطعة القادمة من المرسل لتشغيل برنامج خدمة مقاطعة فى المرسل لإرسال الحرف الجديد بنفس الخطوات السابقة حتى نهاية الإرسال حيث يمكن للمرسل أن ينهى عملية الإرسال عن طريق وضع 1 فى البت 3 من مسجل التحكم فتصبح صفرا على الطرف DSL كدليل على إنهاء عملية الإرسال .
- يرد المستقبل بوضع صفر فى البت 2 من مسجل التحكم التى تخرج صفر أيضا إلى الطرف BSY فى مسجل الحالة فى المرسل لتخبره أن الطرف الآخر مشغول .



شكل (11-11) توصيل جهازى حاسب من خلال مخرج التوازى على كل منهما

الخطوات السابقة أوضحت عملية إرسال البيانات من المرسل إلى المستقبل باستخدام طرف المقاطعة IRQ فى كل من المخرجين . يمكن استخدام إشارات التحكم الأخرى بدلا من ذلك أيضا . كما رأينا فإن طريقة تبادل المعلومات بين جهازى الحاسب كما شرحت سابقا ستعتمد بدرجة كبيرة على مهارة المبرمج ومقدرته فى تقليل وقت المصافحة حيث يمكن بذلك أن يصل إلى معدلات إرسال للبيانات عالية تصل إلى 300 كيلوبايت/ثانية . كما رأينا فإنه من مميزات هذه الطريقة السرعة ، ولكن من عيوبها أنها غير مناسبة للمسافات الطويلة ، فيجب ألا تتعدى مسافة الإرسال 5 متر على الأكثر .

obeikandl.com

5-11 تمارين

1- أكتب برنامج وارسم الدائرة اللازمة لإضاءة 8 لمبات على التوالى بحيث تضىء الأولى لمدة ثانية ثم تطفىء وبعدها مباشرة تضىء الثانية لمدة ثانية ثم تطفىء وهكذا حتى اللمبة الأخيرة حيث بعدها نبدأ من اللمبة الأولى ويتكرر البرنامج إلى أن نضرب أى مفتاح .

2- أعد السؤال السابق ولكن فى هذه المرة نريد إضاءة اللمبة الأولى وبعدها بواحد ثانية تضىء الثانية وهكذا إلى أن تضىء الثمان لمبات مع بعضهم ، بعدها نبدأ فى إطفاؤهم بالعكس ، وهكذا يستمر البرنامج إلى أن نضرب أى مفتاح .

3- أكتب برنامج يشغل المظهر ذو السبع قطع عن طريق توصيل كل قطعة على أحد خطوط البيانات الخارجة من المخرج المتوازى . اختبر البرنامج عن طريق إخراج أرقام معينة ولاحظها على المظهر .

4- وصل مفتاح على أحد خطوط الحالة وليكن الخط ACK ، مع توصيل المظهر ذو السبع قطع كما فى تمرين 3 ، واكتب برنامج يعد عدد مرات ضرب المفتاح وإظهارها على المظهر .

5- نفذ توصيلة جهازى الحاسب مع بعضهما من خلال المخرج المتوازى واكتب برنامج الإرسال والاستقبال بينهما باستخدام خطوط المصافحة وليس بالمقاطعة حيث أنه لم يتم شرح المقاطعة بالتفصيل هنا .

obeikandl.com