

الوحدة الثالثة

برمجة الطبقات الفيزيائية

Physical Layer Programming

2. المنافذ التفرعية Parallel Ports

- يمكننا ببساطة القول أن أرقام المنافذ بين 1 و 65535 هي عبارة عن عناوين تحددها خطوط عنونة المعالج (Address bus)، مخصصة لربطه (أي المعالج) مع التجهيزات المحيطية من أجل تبادل البيانات (وبالتالي المعلومات) عبر خطوط البيانات (Data Bus) المرتبطة مع هذه التجهيزات.
- تتصل الحواسيب مع الوسط الخارجي من خلال المنافذ التي تشكل المنافذ الأساسية للحاسوب، ويستخدم الحاسوب الوصلات التفرعية بشكل أساسي لكي يتمكن من:
- الاتصال مع التجهيزات المحيطية مثل الطابعة printer
- الاتصال مع الوحدات المحيطية أو الحواسيب الأخرى سواء عبر شبكة أو عبر الربط المباشر.
- تلقى المنافذ التفرعية اهتماماً واسعاً في وصلات الحاسوب التي تحتاج إلى سرعة نقل بيانات، ويعتبر الوصل مع الطابعة عبر المخرج LPT1 أشهر استعمالات المنافذ التفرعية.
- والمنافذ التفرعية بشكل عام سهلة الاستخدام وتقدم سرعة نقل بيانات كبيرة إذا ما قورنت بالمنافذ التسلسلية إلا أنها تحتاج إلى عدد كبير من الأسلاك وهذا ما يجعلها نادرة الاستعمال في نقل البيانات لمسافات كبيرة، وبالتالي نادرة الاستخدام في التشبيك.

• 1.2 طرق التبادل عبر المنافذ التفرعية

- تختلف طرق تبادل البيانات عبر المنافذ التفرعية باختلاف طرق استخدامها. ويمكننا تصنيف طرق تبادل البيانات التفرعية تحت أربع عناوين رئيسية هي:

• 1.1.2 الإدخال والإخراج البسيطين Simple Input/Output

- يستخدم الإدخال والإخراج البسيطين حيث نحتاج:
- لتوفر البيانات بشكل دائم على المداخل
- ولا تظهر الحاجة لأي خط تحكمي لإتمام هذه العمليات كما يبين الشكل (1) حيث ترصد تغيرات المعطيات ويتم استقبالها على خطوط البيانات

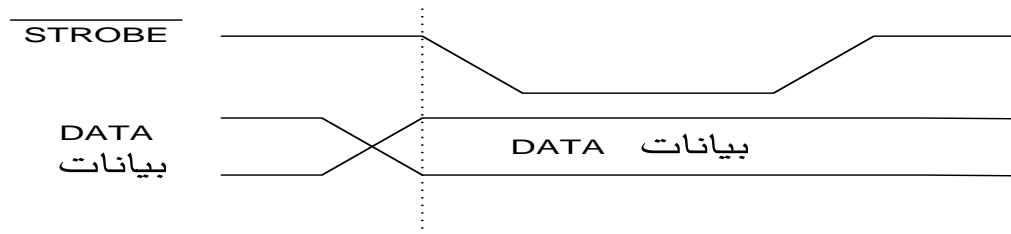
• الشكل (1) الإدخال /الإخراج البسيطين



2.1.2 الإدخال/الإخراج النبضي البسيط Simple Strobe I/O

يستخدم الإدخال/الإخراج النبضي البسيط حين تظهر البيانات التي نرغب بتبادلها في لحظة معينة

كما يمثل الشكل (2) الذي يبين نبضة STROB تنتقل إلى المستوي المنطقي المنخفض (المستوي الفعال) بعد توفر البيانات على خطوط البيانات.

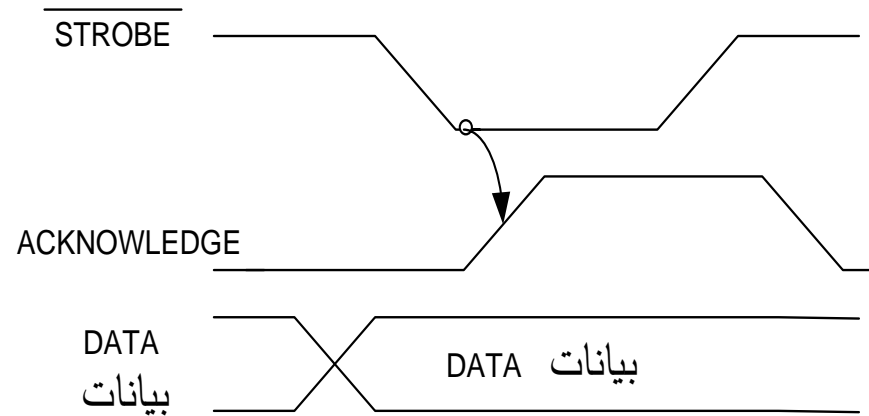


الشكل (2) الإدخال/الإخراج البسيطين

3.1.2 تبادل البيانات باستخدام المصافحة الأحادية Single Handshake I/O

- تستخدم المصافحة الأحادية عندما يكون هنالك حاجة لضمان وصول البيانات، ويحصل ذلك عندما تكون سرعة الإرسال أعلى من سرعة استقبال البيانات.
- يقوم المرسل بإرسال البيانات مرفقةً بنبضة STROBE للدلالة على جاهزيتها على خطوط البيانات إلا أنه لا يرسل كلمة البيانات التالية إلا بعد أن يتلقى من المستقبل إشارة (ACKNOWLEDGE) التي تشعره بأنه قد تمت عملية استقبال هذه البيانات كما يبيت الشكل (3) عندها يستطيع إرسال كلمة البيانات التالية.

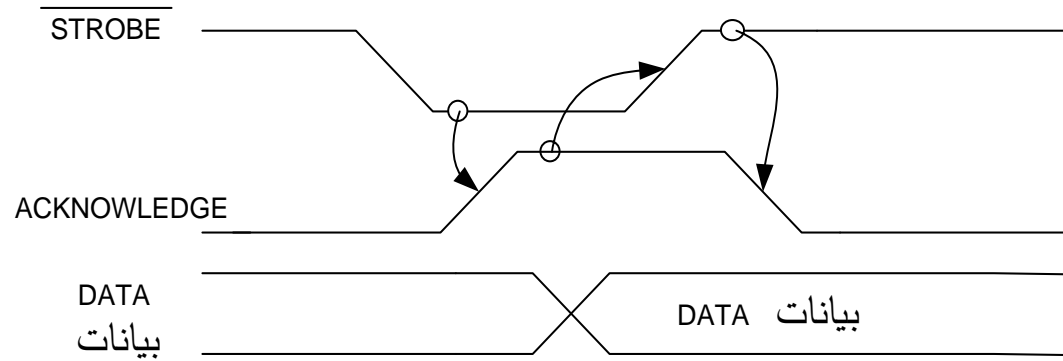
• الشكل (3) المصافحة الأحادية



• 4.1.2 تبادل البيانات باستخدام المصافحة المزدوجة :Double Handshake I/O

- تستخدم المصافحة المزدوجة من أجل ضمان تبادل البيانات على جانبي الإرسال إذ عندها
- يجب على المرسل أن يرسل بياناته عند جاهزية المستقبل لاستقبالها (استفسار عن الجاهزية).
- فيرد المستقبل معلناً جاهزيته للاستقبال.
- يقوم المستقبل باستقبال البيانات مرسلاً إشعاراً بإتمامه عملية الاستقبال.
- ويبين الشكل (4) المصافحة المزدوجة

• الشكل (4) المصافحة المزدوجة

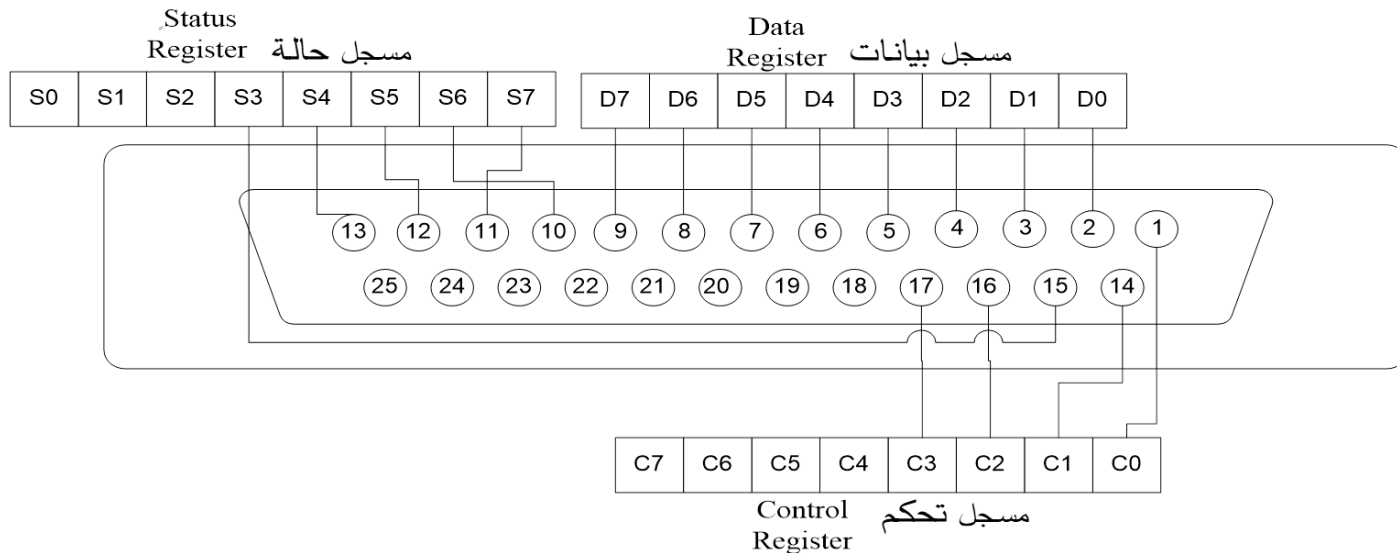


- **2.2 بنية منافذ LPTx والغاوين المخصصة لها**
- إن أكثر استخدامات المنافذ التفرعية شهرةً —كما نوهنا سابقاً— هو استخداماتها في تبادل البيانات مع الطابعة **printer**
- يمكن ملاحظته بسهولة على اللوحة الخلفية **rare panel** للحاسوب الشخصي فهو موصل ذو 25 نهاية أنثوية على شكل حرف **D (DB25)** وهو الموصل الذي نربطه بالطابعة.
- يبين الشكل (5) وظائف النهايات لوصلة اللوحة الخلفية للحاسوب ويبين الجدول (1) وظائف هذه النهايات. ونلاحظ تقسيم نهايات هذا الموصل إلى ثلاث مجموعات وفقاً لوظيفتها وهي:
- 1. نهايات البيانات Data lines** التي توصل عادةً إلى باص البيانات **Data Bus** ونستطيع عادةً رصد تغيراتها في مسجل البيانات **Data Register** في وحدات الإدخال الإخراج. ويجري عبر هذه النهايات تبادل البيانات بين الجهاز المربوط مثل الطابعة والحاسوب.
- 2. نهايات التحكم Control Lines** التي يتم من خلالها التحكم بالأجهزة المحيطة كتفعيل عمليات القراءة **read** أو الكتابة **write**

3. خطوط الحالة Status Lines التي تدل على حالة البيانات مثل توفر البيانات، لحظة ظهورها ولحظة تبادلها وهي مجموعة الإشارات Signals التي يطلق عليها تسميات بحسب نمط التشغيل وترتبط هذه الإشارات مع مسجل الحالة Status

Register

• الشكل (5): نهايات الموصل التفرعي



2.2.2 أنماط المنافذ التفرعية

- تعرّف التقييسات عدة أنماط للتعامل مع البيانات تختلف بحسب الاستخدام والتقنية المتبعة للوصول التفرعي وقد اعتمدت لغة جافا في تعاملها مع الربط التفرعي تقييس IEEE 1284 الذي يعرّف خمسة أنماط رئيسية للربط هي:

1. نمط التوافق Compatibility mode: وهو النمط الذي يعرّف البروتوكول المستخدم من قبل معظم الحواسيب الشخصية لنقل البيانات إلى الطابعة

2. نمط الخانات الأربع Nibble mode: يعتبر هذا النمط الأكثر استخداماً للحصول على البيانات الراجعة من الجهاز المرتبط مع الحاسوب الشخصي مثل الطابعة أو أي جهاز محيطي آخر.

3. نمط الخانات الثمانية Byte mode: ترافق ظهور هذا النمط مع السعي إلى تطوير أداء المنافذ التفرعية وخصوصاً من قبل شركة IBM التي قدمت منفذ PS2 التفرعية .

4. نمط المنفذ التفرعية المحسن Enhanced Parallel Port Mode EPP: جرى تطوير هذا النمط من قبل كل من Intel و Xircom و Zenith لكي يقدم منافذ متوازية عالية الكفاءة متوافقة مع النمط التوافقي.

5. نمط المنفذ مطور الكفاءة Extended Capability Port Mode ECP: قدمت شركتي Hewlett Packard و Microsoft هذا النمط ليكون نمطاً متطوراً يمكن من الاتصال بكل من الطابعة Printer والماسحة Scanner وليكون نمطاً ذا كفاءة عالية يؤمن معبراً ثنائي الاتجاه للبيانات بين المضيف والجهاز المحيطي Peripheral.

- يقدم نمط ECP نوعين من الدورات Cycle types بالاتجاهين الأمامي والعكسي (من وإلى المضيف) وهاتين الدورتين هما:

1. دورة البيانات Data Cycle.

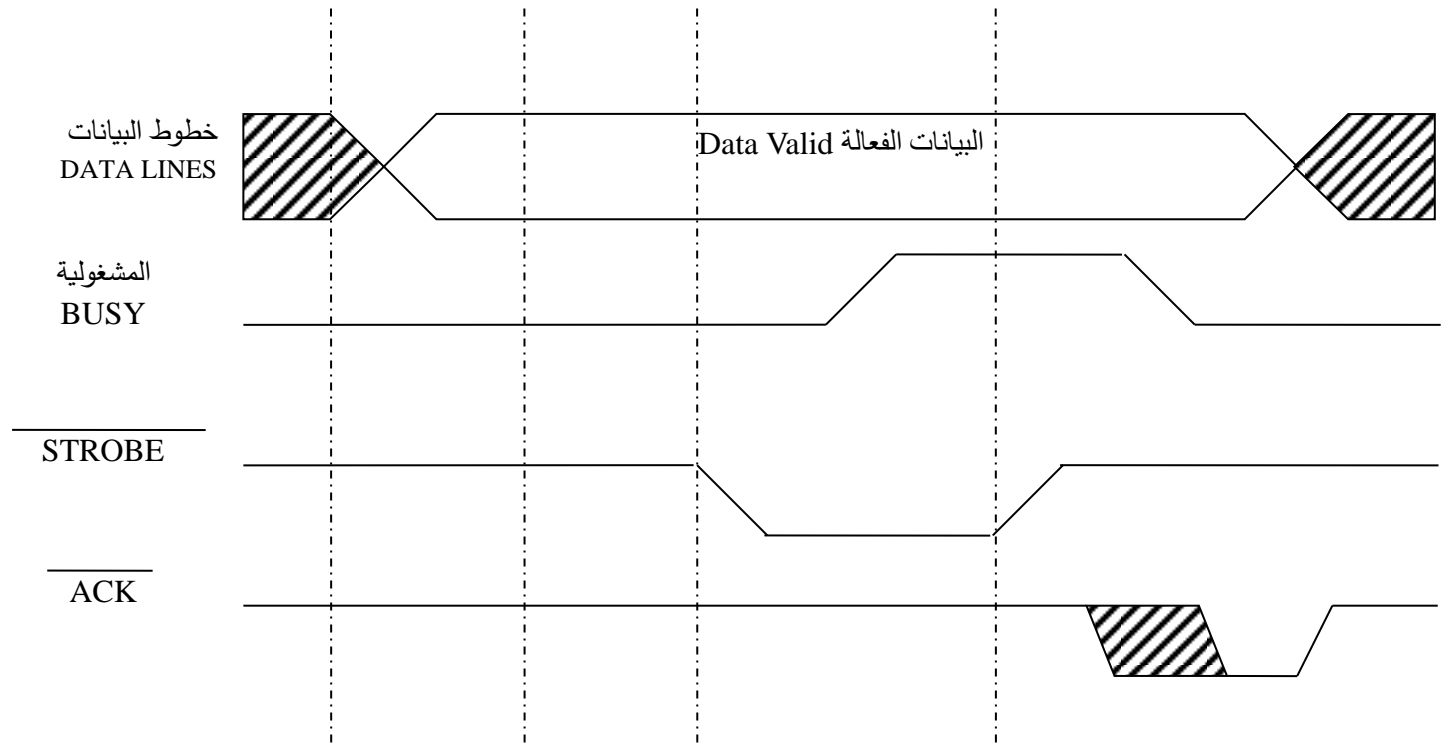
2. دورة الأمر Command Cycle: التي تتعامل بدورها مع رقم يحدد طول التشغيل Run-Length count وعنوان القنال Channel address.

- لقد أخذ بروتوكول ECP التعامل مع مسجلات المنافذ التفرعية بعين الاعتبار وقدم إمكانية ترميز طول التشغيل Run-Length Encoding (RLE)

3.2 خوارزمية نقل البيانات للنمط التوافقي

- ، والنمط التوافقي يقدم طريقة لربط المنفذ القياسية التفرعية للحاسوب بالتجهيزات المحيطية مثل الطابعة.

- الشكل (6): دورة نقل البيانات في النمط₁ التوافقي



- يمر تبادل البيانات عبر المنفذ التفرعية باستخدام هذا النمط بأربع أطوار يبين تتاليها الزمني الأرقام من 1 إلى 4 على الشكل (6) والتي توافق مايلي:
- الطور الأول (1) حيث تكتب البيانات إلى مسجل البيانات.
- الطور الثاني (2) حيث يقرأ البرنامج مسجل الحالة لأجل اختبار مشغولية التجهيزة المربوطة BUSY.
- الطور الثالث (3) الذي يكتب فيه الحاسوب في مسجل الحالة الكلمة التي تضمن تفعيل إشارة توفر البيانات STROBE في حال عدم المشغولية
- الطور الرابع (4) الذي يحرر نهاية STROBE من خلال الكتابة إلى مسجل التحكم.
- ولتطبيق عمليات التبادل فإننا سنتجه لاعتبارات برمجية تتمحور حول قدرات لغة جافا على تطبيق فعاليات تبادل البيانات عبر المنافذ التي تشكل المنفذ الوحيد للحاسوب على الوسط الفيزيائي.

4.2 برمجة المنافذ

- رغم أن معظم لغات البرمجة تسمح للمبرمجين بالوصول إلى المنافذ التفرعية من خلال توابع Functions ضمن مكتباتها القياسية، فإن لغة جافا لا تمتلك هذه الإمكانية لذلك كان لا بد من الاستعانة بمكتبات الربط الديناميكية Dynamic Link Library(dll) المكتوبة بلغات تسمح بهذه الإمكانات مثل C أو C++، تتصل برمجيات جافا مع المنافذ من خلال مجاري الإدخال الإخراج القياسية. وجميع المنافذ هي امتداد لصنف مجرد يسمى commPort هو الصنف الأساس لأي منفذ يمكن استخدامها. وتنقسم الفعاليات التي يقدمها هذا الصنف إلى قسمين:
- فهو يسمح بفتح وإغلاق والقراءة من أو الكتابة في أي من المنافذ.

الجدول (3) مكونات الحزمة البرمجية java.comm

الاستثناءات Exceptions	الواجهات Interfaces	الأصناف Classes
NoSuchPortException	CommDriver	CommPort
PortInUseException	CommPortOwnershipListener	CommPortIdentifier
UnsupportedCommOperationException	ParallelPortEventListener	ParallelPort
	SerialPortEventListener	ParallelPortEvent
		SerialPort
		¹⁷ SerialPortEvent

تلخيص وظائف هذه العناصر البرمجية كما يلي:

- **الصف comPort :** هو كما ذكرنا صف مجرد أساس لجميع المنافذ يمكننا من التعامل والتحكم بجميع المنافذ بنفس الطريقة ويمتلك جميع الطرق التي تسمح لنا بالتعامل مع المنافذ مثل الطرق `getOutputStream()` و `getInputStream()` التي تسمح بالقراءة والكتابة في المنفذ إضافةً إلى الطريقة `close()` التي تغلق المنفذ.
- **الواجهة CommDriver** التي تمنحنا إمكانية إضافة أشكال اتصالات جديدة للاتصال من خلال إضافة منافذ جديدة بحسب حاجة المستخدم.

- وتظهر الحاجة إلى بعض الطرق التي نستفيد منها في برمجياتنا مثل

- الطريقة `public static Enumeration getPortIdentifiers()` التي تقدم لنا قائمة بجميع المنافذ في النظام تسلسليةً كانت أو تفرعية.

- الطريقة `public String getName()` التي تعطينا اسم معرف المنفذ.

- الطريقة `public int getPortType()` التي تعطينا فيما إذا كان معرف المنفذ ينتمي إلى:

`CommPortIdentifier.PORT_SERIAL` أو إلى
`CommPortIdentifier.PORT_PARALLEL`

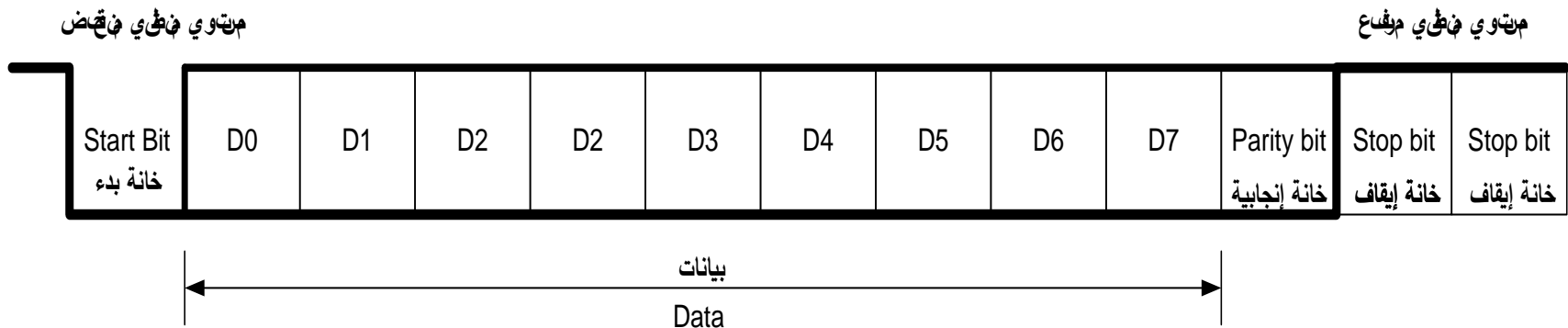
مثال 1: على الطالب +النظر

3. المنافذ التسلسلية Serial Ports

- تعتبر المنافذ التسلسلية المنافذ الأساسية لرسائل البيانات المتبادلة عبر الشبكات
- سيتركز اهتمامنا هنا على آليات نقل البيانات عبر الموصلات التسلسلية بعد تحويلها إلى سلسلة من الخانات الثنائية والذي يجري بطرق ثلاث بحسب الحاجة التقنية وهي:
 - **نقل البيانات البسيط وحيد الاتجاه Simplex:**
 - **نقل البيانات النصفى Half Duplex:**
 - **نقل البيانات التام ثنائي الاتجاه Full Duplex:**
- ويمكننا تصنيف نقل البيانات التسلسلي وفقاً لتوقيت الإرسال والاستقبال فيما يطلق عليه التراسل المتزامن Synchronous والتراسل غير المتزامن Asynchronous.

1.3 التراسل غير المتزامن Asynchronous

- الشكل (7) الشكل العام للرمز التسلسلي في التراسل غير المتزامن
- حيث نلاحظ إضافة خانات التحكم التالية:
- خانة البدء Start bit (مستوي منطقي منخفض).
- خانة الإنجابية Parity bit
- خانتي إيقاف Stop bit (خانة، خانة ونصف أو خانتين).
- وترسم هذه الخانات حدود كل كلمة معطيات متبادلة مع إمكانية كشف الخطأ عبر خانة الإنجابية Parity bit.



2.3 التراسل المتزامن Synchronous

- يتميز التراسل المتزامن:
- بمعايرته وتنظيمه طرفي المرسل والمستقبل وإعدادهما لعملية تبادل البيانات قبل الشروع بعملية التراسل.
- تبادله كتلة Block كبيرة نسبياً من البيانات دفعة واحدة لنفس مجموعة إشارات التحكم.
- وأشهر بروتوكولات تبادل البيانات التسلسلية المتزامنة هي:
- بروتوكول الاتصال المتزامن الثنائي Binary Synchronous Communication Protocol .BSYNC
- بروتوكول التحكم بربط البيانات عالي المستوى High Level Data Link Control HDLC .

3.3 مصطلحات النقل التسلسلي

- تعرفنا في المقررات السابقة على الجهاز المعدل- كاشف التعديل MODEM والمعدات الأخرى التي تستخدم لنقل البيانات لمسافات كبيرة والتي أطلق عليها اسم معدات اتصال البيانات Data Communication Equipment DCE، بينما سمي الحاسوب الذي يرسل أو يستقبل البيانات بالمعدات الطرفية للبيانات Data Terminal Equipment DTE. ويبين الشكل (8)
- تقوم DTE بعد وصلها تشغيلها بتفعيل إشارة جاهزية المعدات الطرفية (DTR Data Terminal Ready) التي تبلغ المودم بجاهزيتها، فإذا كان المودم في حالة الجاهزية لتبادل البيانات فإنه يبلغ عن جاهزيته من خلال إشارة جاهزية مجموعة البيانات DSR Data Set Ready التي ترسل إلى الحاسوب وبذلك تتم مرحلة إنشاء الاتصال.

الشكل (8): الإشارات التي تتبادلها معدات اتصال البيانات مع
المعدات الطرفية للبيانات

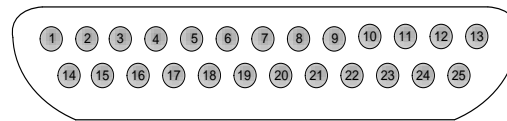
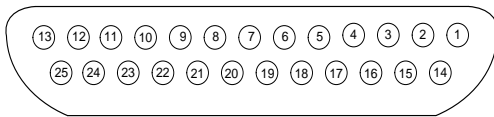
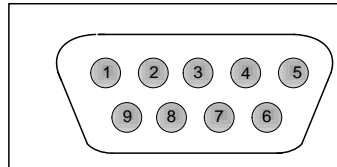
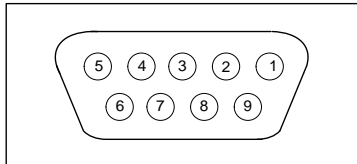
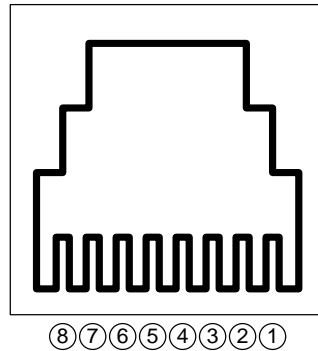
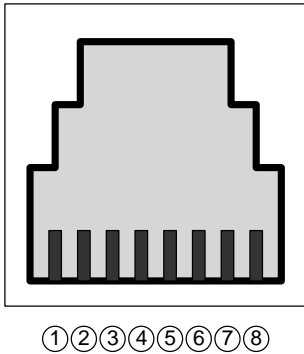
• 111 page

1.3.3 تقويس RS232

- ويوضح الشكل (8) الموصلات المستخدمة للوصل التسلسلي: موصل

- 9-Pin موصل RJ45

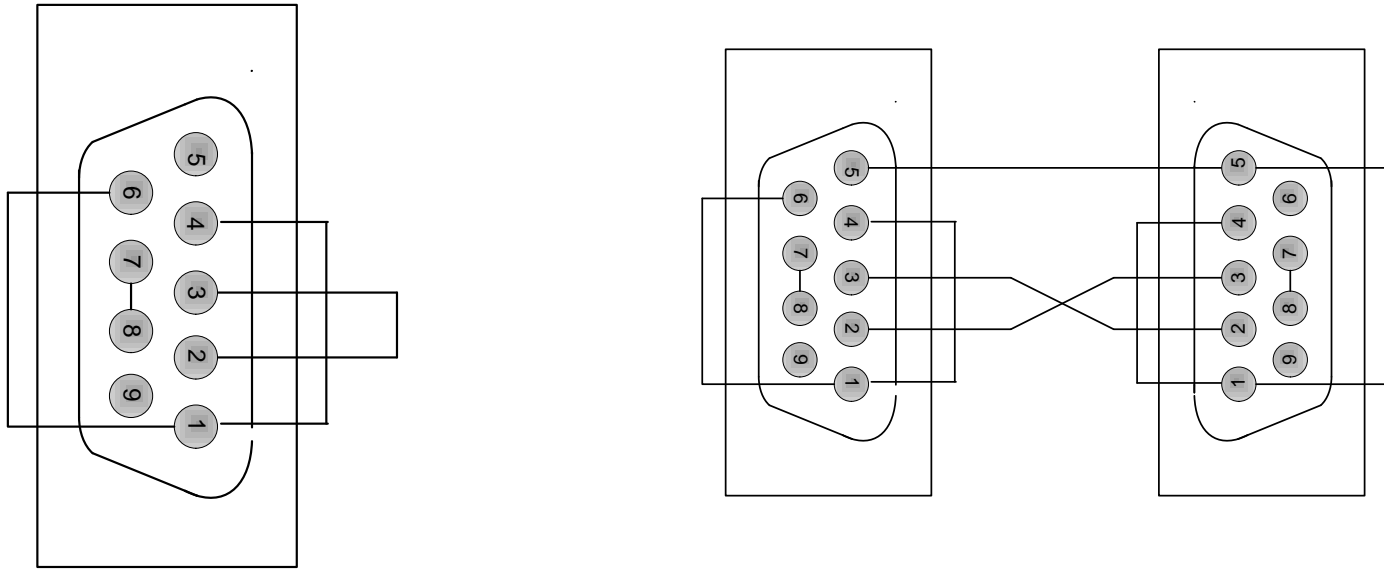
- 25-Pin موصل



4.3 تصميم وسط نقل البيانات

- يرتبط تصميم وسط نقل البيانات التسلسلي مع البروتوكول المستخدم (وهنا هو RS232) من جهة ومع شكل الموصل من جهة أخرى (وهنا هي الموصلات عبر RJ45 و موصلات النهايات التسع 9-Pin connectors والموصلات ذات النهايات الخمس والعشرون 25-pin connectors).
- ونميز بالنسبة لوسط النقل بين الموصلات الارتباطات التالية:
- وصلة DTE إلى DCE التي تتم عبر الوصل المباشر.
- وصلة DTE إلى DTE التي تتم عبر المودم الصفري.
- وصلة DCE إلى DCE التي تتم عبر كابلات الدوائر الخلفية Tail Circuit Cable.

وسنوضح على سبيل المثال لا الحصر من خلال الشكل (10) الذي يمثل وصلة
المودم الصفري والشكل (11) الذي يمثل الحلقة الراجعة. ويمكننا استخدام كلتا
الوصلتين في تطبيقاتنا من أجل ضمان وصول البيانات. عبر منافذ COMx.



• 5.3 التنفيذ العملي لبرنامج يوضح عملية نقل البيانات

- سنقدم فيما يلي برنامجاً يمكننا من نقل البيانات عبر المنفذ التسلسلية حيث يهتم البرنامج الأول بعملية القراءة من المنفذ التسلسلية بينما يهتم المثال الثاني بعملية قراءة البيانات منها.

• برنامج القراءة:

- يستخدم هذا البرنامج واجهتي *Runnable* و *SerialPortEventListener* وهذا يعني أنه من الواجب علينا انتظار النداء الراجع *callback* من أحداث معالجات المنافذ التسلسلية *SerialEvents() Event Handler*. كما أننا سنستخدم المسالك في تطبيقنا هذا.

```

import java.io.*;
import java.util.*;
import javax.comm.*;

public class SimpleRead implements Runnable, SerialPortEventListener
{
    static CommPortIdentifier portId;
    static Enumeration portList;
    InputStream inputStream;
    SerialPort serialPort;
    Thread readThread;

    public static void main(String[] args)
    {
        boolean portFound = false;
        String defaultPort = "COM1";

        if (args.length > 0) {
            defaultPort = args[0];
        }

        portList = CommPortIdentifier.getPortIdentifiers();

        while (portList.hasMoreElements()) {
            portId = (CommPortIdentifier) portList.nextElement();
            if (portId.getPortType() == CommPortIdentifier.PORT_SERIAL) {
                if (portId.getName().equals(defaultPort)) {
                    System.out.println("Found port: "+defaultPort);
                    portFound = true;
                    SimpleRead reader = new SimpleRead();
                }
            }
        }

        if (!portFound) {

```

```

System.out.println("port " + defaultPort + " not found.");
    }
}

public SimpleRead() {
    try {
        serialPort = (SerialPort) portId.open("SimpleReadApp", 2000);

    }
    catch (PortInUseException e) {}
    try {
        inputStream = serialPort.getInputStream();
    }
    catch (IOException e) {}
    try {
        serialPort.addEventListener(this);
    }
    catch (TooManyListenersException e) {}

    serialPort.notifyOnDataAvailable(true);

    try {
        serialPort.setSerialPortParams(2400, SerialPort.DATABITS_8,
            SerialPort.STOPBITS_1, SerialPort.PARITY_NONE);
    }
    catch (UnsupportedCommOperationException e) {}
    readThread = new Thread(this);
    readThread.start();
}

public void run() {
    try {
        Thread.sleep(20000);
    }
    catch (InterruptedException e) {}
}

```

```

public void serialEvent(SerialPortEvent event) {
    switch (event.getEventType()) {
        case SerialPortEvent.BI:
        case SerialPortEvent.OE:
        case SerialPortEvent.FE:
        case SerialPortEvent.PE:
        case SerialPortEvent.CD:
        case SerialPortEvent.CTS:
        case SerialPortEvent.DSR:
        case SerialPortEvent.RI:
        case SerialPortEvent.OUTPUT_BUFFER_EMPTY:
            break;
        case SerialPortEvent.DATA_AVAILABLE:
            byte[] readBuffer = new byte[20];
            try {
                while (inputStream.available() > 0) {
                    int numBytes = inputStream.read(readBuffer);
                    System.out.print(new String(readBuffer));
                }
            } catch (IOException e) {}
            break;
    }
}

```

المثال (2) برنامج القراءة من منفذ الاتصال التسلسلية

- نلاحظ في `main()` أنه ما لم نحدد المنفذ التي نتعامل معها في سطر الأوامر فإن المنفذ التي سيتعامل مع البرنامج هي `COM1`، إذ أنه باستخدام `CommPortIdentifier` سيقوم البرنامج بإيجاد المنفذ ومن ثم بإنشاء كيان من `SimpleRead`. ويتوجب علينا عندئذٍ أن نفتح المنفذ ونحصل على مجرى الدخل (باستخدام الصنف `IO`) ونسجل كيان `SimpleRead` كمستمع ينتظر حدثاً من المنفذ التسلسلية `serial port event listener`. وهنا علينا أن نبذل المنفذ بالأحداث التي نهتم بها من خلال استدعاء الطريقة `notifyOnDataAvailable()`. وأخيراً نقوم بإنشاء مسلك جديد ونضع الكيان `SimpleRead` في مرحلة قابلية التشغيل (في حالة `Runnable`) ونشغل المسلك.
- ويقرأ البرنامج من خلال حلقة البيانات الواردة مادام هنالك بيانات موجودة باستخدام طريقة مجرى الدخل `available()` التي تشير إلى توفر البيانات على مجرى الدخل.
- وبتحديدنا فترة النوم بعشرين ثانية فإن سنفسر المسلك أن ينهي فعالياته بعد مرور فترة محددة على آخر حدث `DATA_AVAILABLE` تم إرساله.

4. الشبكة الرقمية للخدمات المتكاملة Integrated Services Digital Network ISDN

- تقدم الشبكة الرقمية للخدمات المتكاملة ISDN خدمات الاتصالات الهاتفية إضافةً إلى خدمات نقل البيانات على الخطوط الهاتفية النحاسية وتتميز عن الخدمة التقليدية برقمنة Digitizing البيانات
- 1.4 أجهزة الشبكات الرقمية المتكاملة
- أجهزة الشبكات الرقمية المتكاملة هي:
- طرفيات Terminals
- موائمات الطرفيات TA Terminal adapters،
- أجهزة إنهاء الشبكة Network-termination devices،
- تجهيزات إنهاء الخط Line-Termination equipment،
- تجهيزات إنهاء البدالة Exchange-Termination
- Equipment.

وتصنف طرفيات الشبكات الرقمية المتكاملة ضمن نمطين هما:

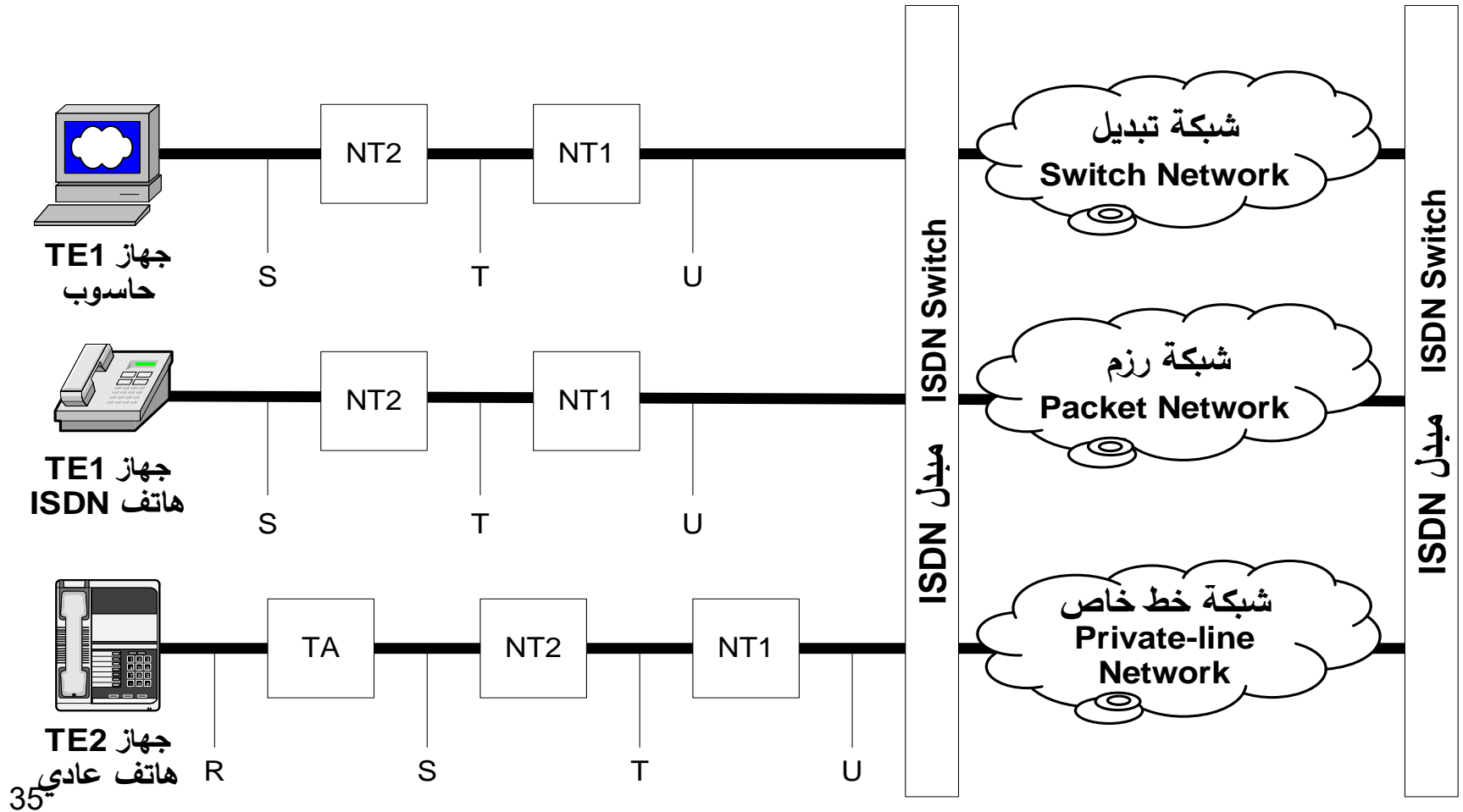
– تجهيزات الطرفيات نمط 1 terminal equipment type 1 (TE1) وهي الطرفيات المخصصة للشبكات الرقمية المتكاملة.

• تجهيزات الطرفيات نمط 2 terminal equipment type 2 (TE2) وهي التجهيزات

ترتبط تجهيزات الطرفيات TE1 بشبكات ISDN بأربعة أسلاك مجدولة ثنائياً twisted pair بينما ترتبط تجهيزات الطرفيات TE2 من خلال موائمات الطرفيات TA. وموائمات الطرفيات هي إما أجهزة مستقلة أو لوحات إلكترونية جرت مكاملتها على لوحة تجهيزات الطرفية TE2.

- قد تم دمج وظائف أجهزة NT1 و NT2 في جهاز واحد.
وتحدد ISDN نقاط مرجعية تعرف الواجهة المنطقية
الفاصلة بين المجموعات الوظيفية المختلفة

• الشكل (12): شبكة ISDN والأجهزة والنقاط المرجعية الخاصة بها



2.4 خدمات ISDN

تتميز شبكات ISDN بنوعين من الخدمات بحسب معدل نقل البيانات والقنوات المستخدمة هما:

• 1.2.4 خدمة الواجهة البينية ذات المعدل الأساسي Basic Rate Interface BRI

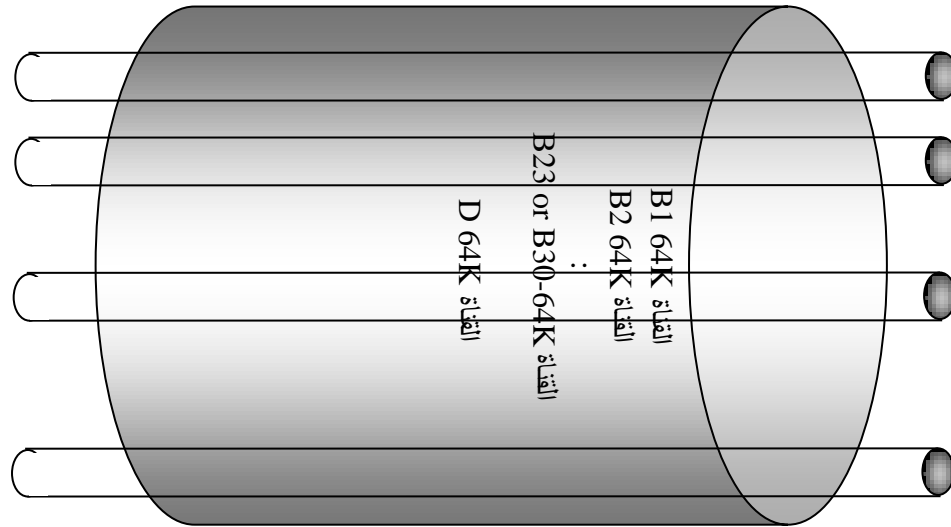
- التي تقدم خدماتها من خلال قناتين من نوع B وقناة من نوع (2B+D). حيث تخصص القناتين **B** لنقل البيانات بسرعة 64 كيلوبت في الثانية، بينما تخصص القناة **D** لنقل الإشارات التحكمية بسرعة 16 كيلوبت في الثانية كما يبين الشكل (13).



الشكل (13) قنوات BRI (2B+D)

2.2.4 خدمة الواجهة البينية ذات المعدل الأولي Primary Rate Interface PRI

- تقدم خدماتها من خلال 23 قناة من نوع B وقناة واحدة من نوع $(23B+D)$. حيث تخصص كل قناة من القنوات B لنقل البيانات بسرعة 64 كيلوبت في الثانية، بينما تخصص القناة D بسرعة 64 كيلوبت في الثانية من أجل الإشارات التحكمية لجميع القنوات B. ويبلغ عدد قنوات B هو 23 قناة في أمريكا الشمالية واليابان أي ما يعادل نسبة نقل بيانات تصل إلى 1.544 ميجا بت لكل ثانية، بينما يصل عدد القنوات في أوروبا وأستراليا وبقية أنحاء العالم إلى 30



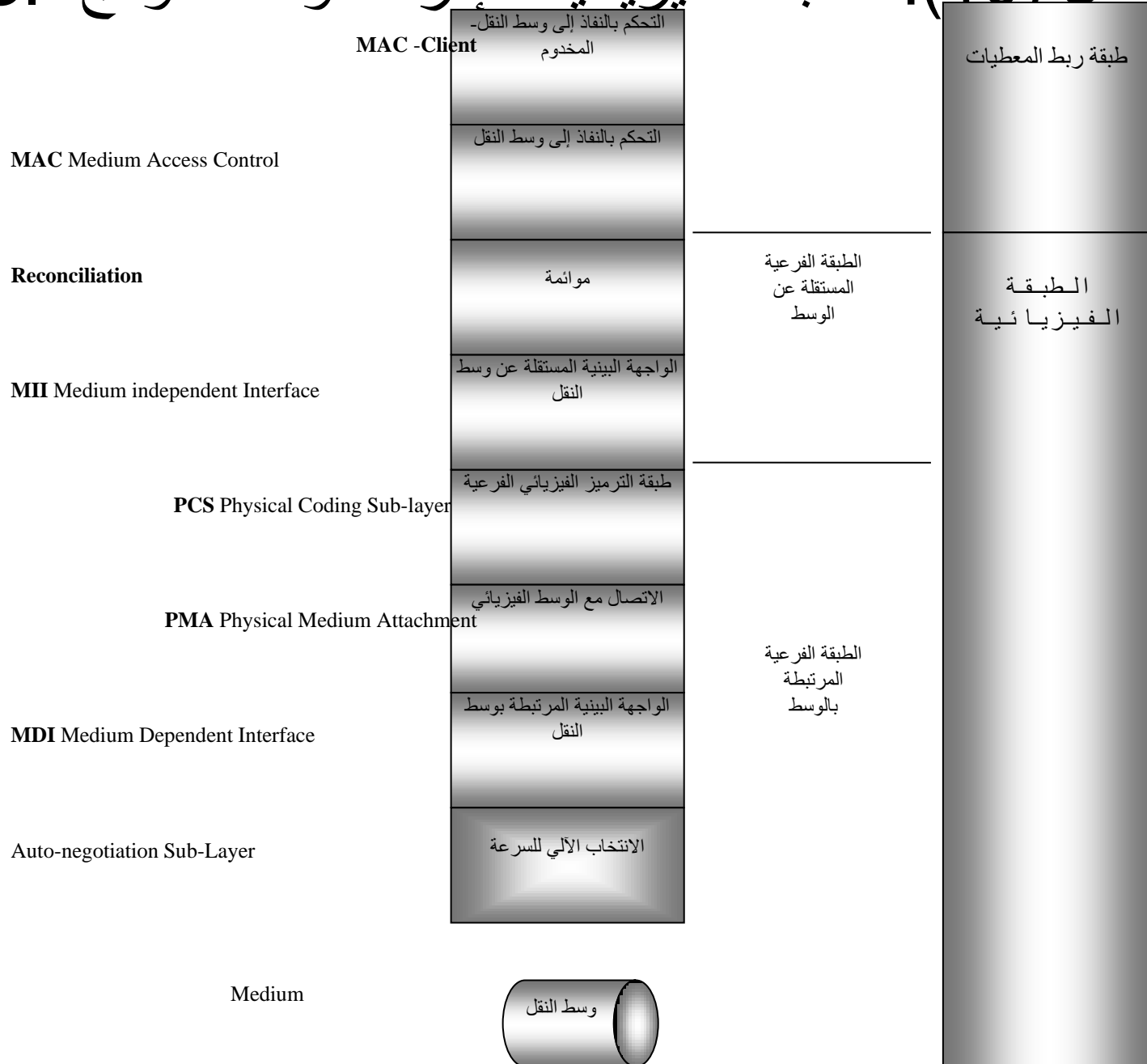
الشكل (14) قنوات PRI
(30 or 23B+D)

- تقسم مهام ISDN على طبقاتٍ ثلاث تختص الطبقة الأولى بالفعاليات الفيزيائية بينما تتضمن الطبقتين 2 و 3 فعاليات الطبقات العليا لنموذج OSI، وسنقتصر هنا على إيضاح فعاليات الطبقة 1 (الفيزيائية) ووظائفها الأساسية.

5. بنية بطاقة الإترنت Ethernet Card

- تغطي تجهيزات وملحقات الإترنت من خلال بطاقة الإترنت وظائف الطبقتين الأولى والثانية -الفيزيائية وربط المعطيات- لنظام التشبيك الداخلي المفتوح OSI.
- تقسم الطبقة الفيزيائية كما يبين الشكل (16) وفقاً لنسبة نقل البيانات Transmission rate إلى طبقات فرعية:
 1. طبقة مستقلة عن وسط النقل
 2. طبقة مرتبطة بوسط النقل وبالتشفير المستخدم للإشارات.

• الشكل (16): الطبقة الفيزيائية للإترنت وفقاً لنموذج OSI



1.5 البنية الوظيفية لبطاقة الشبكة

- كان ارتباط بطاقات الشبكة مع سكة ISA فيما سبق هو الأكثر شيوعاً، إلا أن الانتشار الكبير لسكة PCI وسرعتها الكبيرة بالمقارنة مع سكة ISA سمحت لهذه التقنية باحتلال المكانة الأولى في وقتنا الحالي.
- تتميز بطاقة الشبكة كما يبين الشكل (17) باحتوائها على:
- وحدة التحكم بالنفاز إلى الوسط **MAC**،
- وهي كما نعلم صلة الوصل بين مكونات الطبقة الأولى والثانية من طبقات **OSI**، وهي ركن أساسي في تحقيق البروتوكولات في مستوي الربط من مثل الإترنت،
- النفاز المباشر إلى الذاكرة **DMA**
- الذي يسمح بتبادل البيانات بشكل مباشر بين الذاكرة الرئيسية وذاكرة بطاقة الشبكة مما يزيد من سرعة تبادل البيانات ويقلل من الضغط على المعالج.

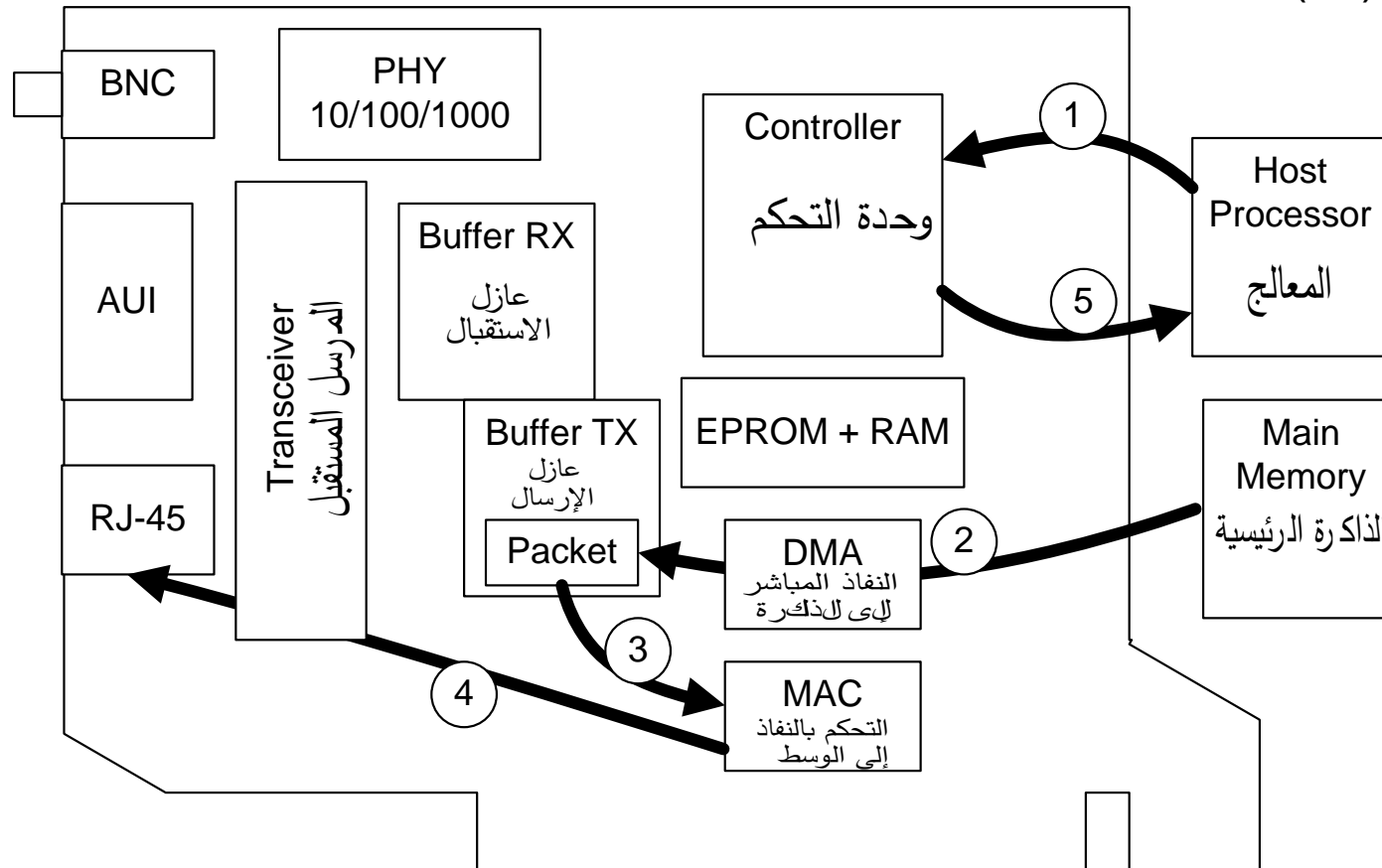
- العتاد الصلب المتعلق بمعالجة الإشارة ضمن الطبقة المسماة

بـ PHY

- ذاكرة الإرسال والاستقبال المؤقتة receive and transmit buffers

- مجموعة الموصلات: حيث أنه رغم سيطرة موصلات RJ45

الشكل (17): إرسال البيانات عبر بطاقة الشبكة



الشكل (18) استقبال البيانات عبر بطاقة الشبكة

