

報 告

マイクロコンピュータインタフェースの
システム開発について (その3)

試作学習支援システムのソフト・ハードウェア制御

Design for Micro Computer Interface System

渋 井 二三男*
SHIBUI, Fumio

概要：前回，研究誌に表記システム開発で欠くべからず Memory Map 設計技術ソフトウェアデータ構造最適設計，ロボット制御，アクチュエート制御，ネット制御に必要な起動制御，ステップ制御，入出力制御方式などについて，その重要性もふくめ，詳細に論述した。

本稿では，プリンタに代表される PIO 制御，調歩同期方式，IBM-SDLC 方式に対応できる SIO 制御……などについて，詳細に論述する。

1. マイコンのファミリー LSI

マイクロコンピュータは，現在あらゆる分野に使用され，マイコンシステムの構成も多様化している。このように多様化するマイコンシステムをそのつど設計するのは大変なことである。そのため，Z80 などの CPU を販売しているメーカーは，CPU と一緒にその CPU に容易に接続して使用できる入出力インタフェース (LSI) を提供し，多様化するマイコンシステムの設計を容易に行なえるようにサポートしている。

この入出力インタフェースを，CPU に対し，常に一緒 (ペア) に使用されることからファミリー LSI と呼んでおり，Z80 CPU では以下に示すファミリー LSI が提供されている。

- PIO (パラレル I/O インターフェース：Parallel Input Output Controller)
- SIO (シリアル I/O インターフェース：Serial Input Output Controller)
- CTC (カウンタ/タイマ：Counter/Timer Circuit)
- DMA (ダイレクトメモリアクセスコントローラ：Direct Memory Access Controller)

* 城西短期大学

2. デコーダ回路

ここでは、デコーダ回路がマイコンシステムの中でどのように使用されているのかについて説明する。

今まで説明の中で、メモリの****番地とか？プリンタにデータを出力する！などの言葉が出てきたが、メモリの特定番地を選択したり、複数の入出力装置の中からプリンタを選択したりするのは、ハードウェアではどのように行なわれているのだろうか？

例で考えてみる。いま図1に示す8kbyteの容量を持つメモリを使用したマイコンシステムについて説明する。

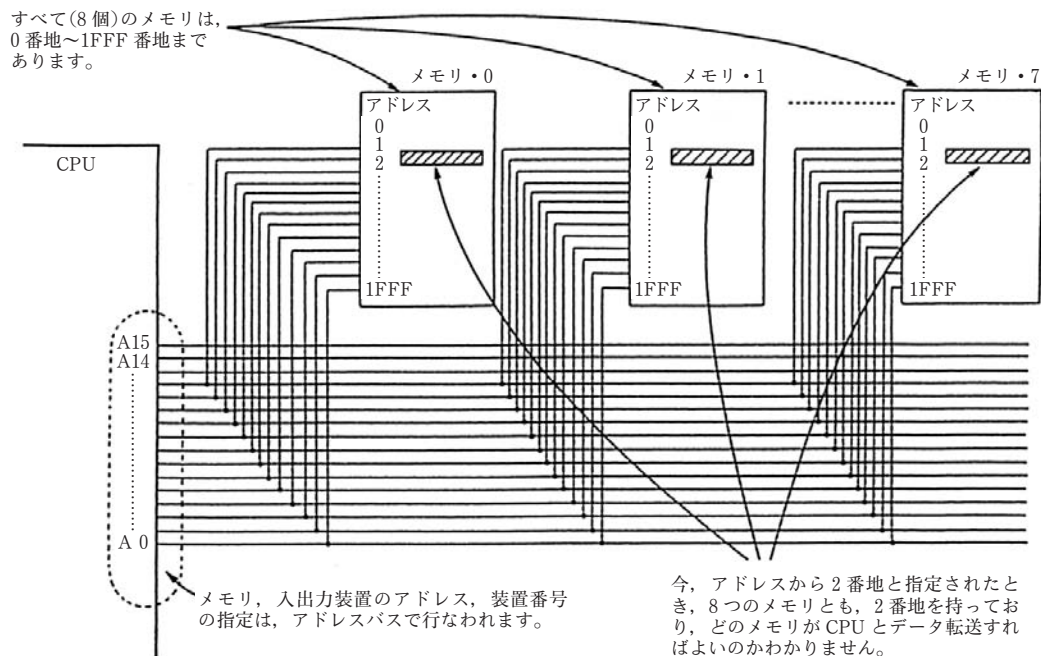


図1 メモリ回路の選択

この図では、2番地が8箇所であり、どのメモリの2番地が本物なのかわからない。そのためにアドレスバスの上位3ビットを使用して、“000”のときはメモリ0，“001”のときはメモリ1，“111”のときはメモリ7というようにメモリを選択してやらなければならない(図2)。

説明では、メモリ0~メモリ7というようにメモリ選択にデコーダ回路が使用されるといったが、各メモリの中においても0番地から1FFF番地までを選択する回路として、このデコーダ回路が

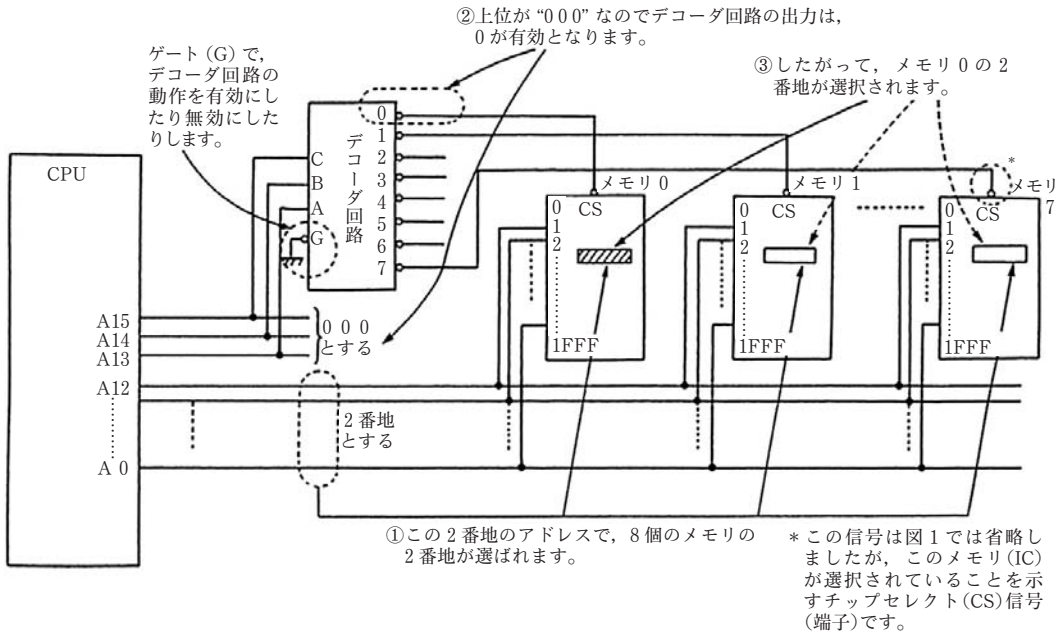


図2 メモリ選択回路

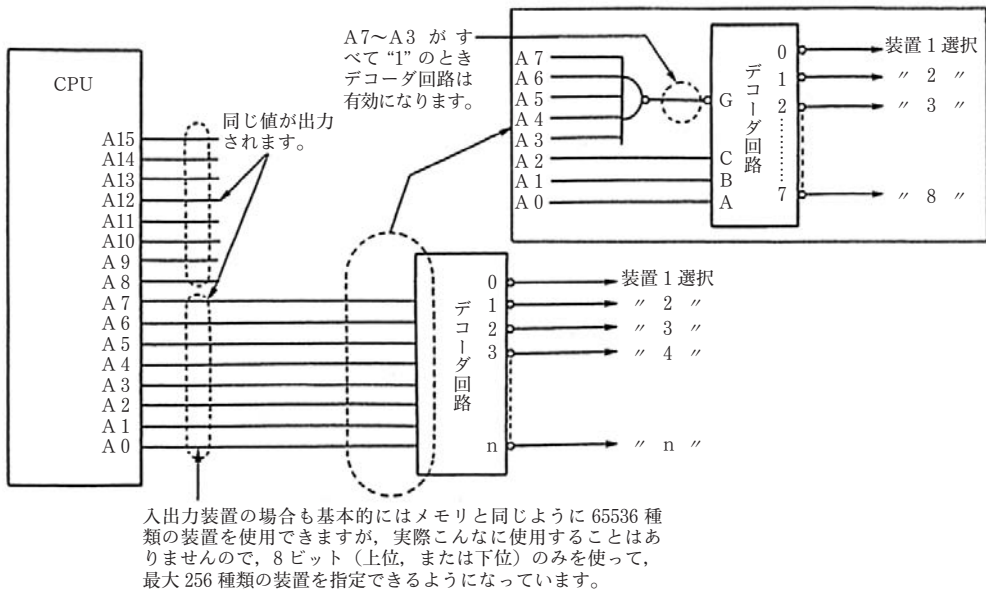


図3 入出力装置の選択

使用されている。

このようにデコーダ回路は、メモリ選択(および、該当アドレスを選択)したり、入出力装置を選択する場合に使用されている。

3. データ転送のタイミング

マイコンシステムに限らずいろいろなシステムで必ずといっていいくらい行なわれるのがこのデータ転送である。ここでは、正しくデータ転送を行なわせるにはどうすればよいかについて説明する。

図4に示す回路で、AからBへデータを転送する方法を考えてみる。

この転送手順は、①Aがデータを出力する、②Bがデータを入力する、と簡単なことだが、次のことが問題になる。

①でAはいつ、どのくらいの時間データを出力しているのか？

②でBはいつデータを入力するのか？

これらの問題を図5で説明する。

転送1では、Aのデータがデータバス上に出力されるより前にBはデータを入力し、転送2では、Aのデータがデータバスに出力し終わってから入力している。したがって、このタイミングで

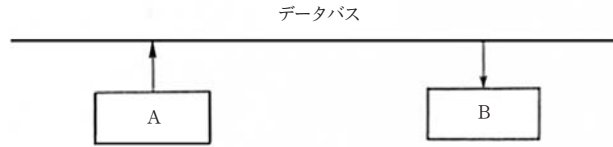


図4 データ転送

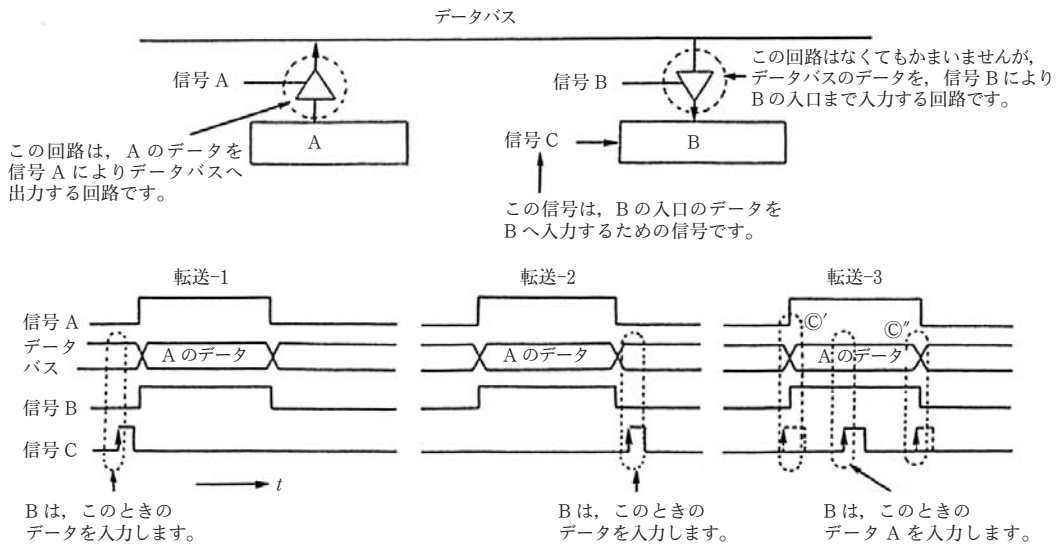


図5 データ転送のタイミング

は、図からもわかるように、データバス上には、A のデータがないので、B には、A のデータが転送されない。

転送3で示すように、A のデータがデータバス上に出力されているあいだに入力しなければならない。ただし、A のデータが出力されている間といっても、図5の③の一では、A のデータがデータバスからなくなり始めているところなので、B へ正しく入力されているとは限らない。したがって、図に示すように、データが出力されている期間の中央（データが安定しているなどという）で入力することが必要である。

4. 信号の種類

今までは、論理回路の AND 回路などで説明してきたように、入力は0と1があり、特に1に注目してきた。

このように、0は信号のない状態（無効状態）、1を信号のある状態（有効状態）としたり、反対に、デコード回路の出力では、1が無効状態、0が有効状態になっていた。

このように、0、1のどちらかを有効状態にするかにより、図6に示すように、正論理（1を有効状態にする）と負論理（0を有効状態にする）に分けられる。

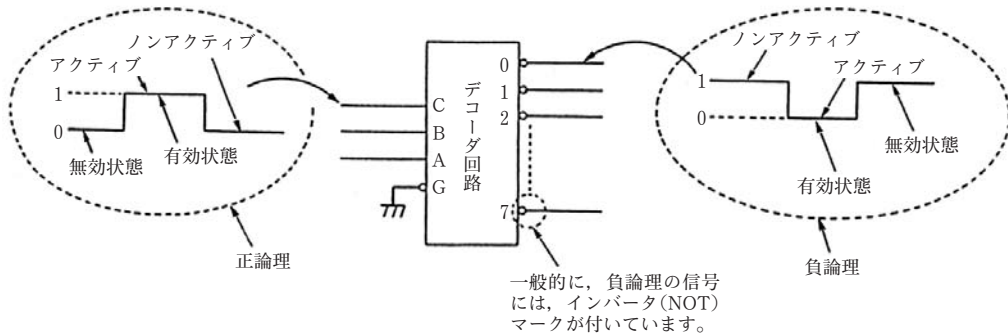


図6 正論理と負論理

また、正論理、負論理にかかわらず、有効状態をアクティブ、無効状態をノンアクティブと呼んでいる。

信号 A がアクティブの間、A のデータ出力、信号 B がアクティブの間、データを B の入り口まで入力し、信号 B で B へ入力していた。

この信号 A、信号 B、信号 C を比べると、図7に示すように、信号 A と信号 B では、アクティブ、またはノンアクティブかによりデータの入出力が制御されているが、信号 C は 0→1 へ変化した状態（瞬間）を使用している。

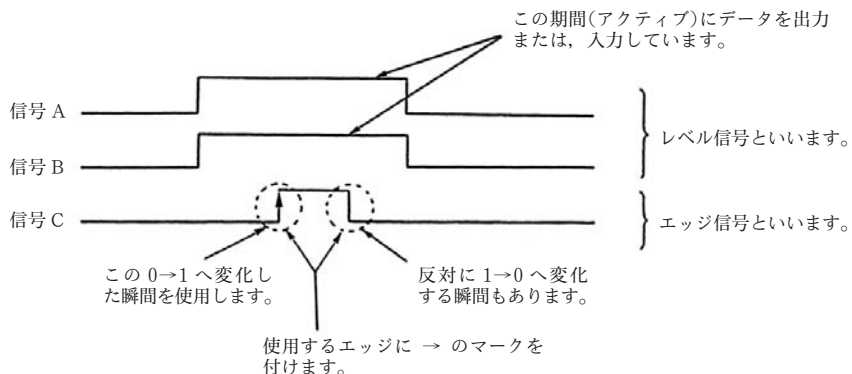


図7 レベル信号とエッジ信号

このように0→1(または、反対の1→0)へ変化する瞬間を利用する信号をエッジ(桁を利用するのでエッジと呼ぶ)信号といい、0→1へ変換する信号を立ち上がりエッジ信号、1→0へ変化する信号を立下りエッジ信号と呼んでいる。

これに対し、信号A、信号Bはレベル信号と呼んでいる。

5. 標準 LSI

5.1 パラレルインタフェースの動作

次にZ80 CPU 応用システムとして最も代表的なZ80 ファミリの代表的なインタフェースについて説明する。

Z80 PIO はデータの並列処理を行うための周辺 LSI で、プリンタ、スイッチ、ランプ、各種ステータス等を CPU との間で並列にデータ転送を行うためのインタフェースに使用される。

Z80 PIO (Parallel Input/Output Controller : PIO) は、8 ビット入出力ポートが2つ内蔵されており、4本のハンドシェイク用制御信号線がある。

各ポートは8 bit・パラソルのみでなく、1ビットごとに入力または出力としてプログラムで設定できるようにもなっている。

さらに、各ポートに任意の割込み用ベクトルをもつことができる。

PIO のブロック図を図8に示す。

では、PIO の動作を簡単に説明する。PIO の動作には、次の4種類の動作モードがある。

- モード0 (出力モード)
- モード1 (入力モード)
- モード2 (双方向モード)
- モード3 (ビットモード)

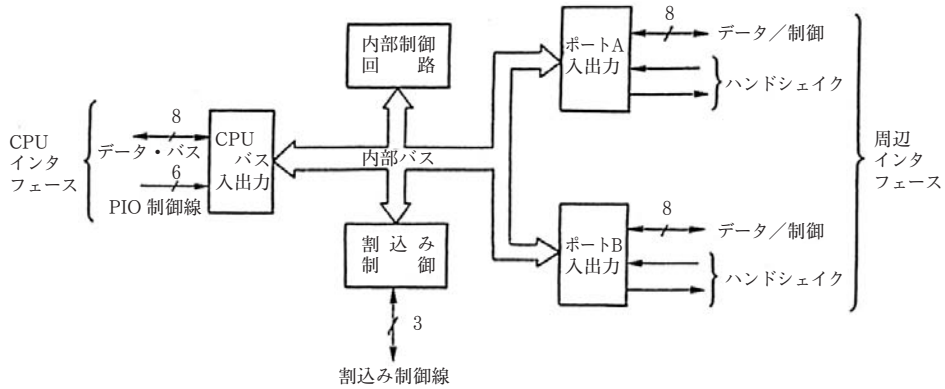


図8 PIOのブロック図

(1) モード0の動作

このモードでは、ポートA、またはポートBを出力ポートとして使用する方で、コンピュータから出力されたデータをポート内にラッチすることができる。信号の動作としては、コンピュータからデータが出力されるとポート内にラッチされ、データがP*7~P*0へ出力され、出力データを出力装置へ出力したことを知らせるために*RDY信号を“H”にする。

その後、出力装置がデータを受け取り、*STB信号“L”にすると*RDY信号は“L”に戻る。また、このとき割込み許可状態であれば、コンピュータに出力終了を知らせるための割込みが発生する。

図9に使用例を示す。

(2) モード1の動作

このモードでは、ポートA、ポートBを入力ポートとして使用する方で、入力装置からのデータをポート内にラッチし、コンピュータへ出力することができる。信号の動作としては、入力装置がP*7~P*0へデータを出し、その後*STB信号を“L”にするとポート内バッファへデータがラッチされ、*RDY信号はポート内にデータがラッチされていることを知らせるため“L”になる。

このとき、割込み許可状態であれば、コンピュータにデータの準備が完了したことを知らせるための割込みを発生し、コンピュータが入力命令でデータを読み取ると*RDY信号を“H”にし、入力装置へ次のデータを要求する。

図10に動作例を示す。

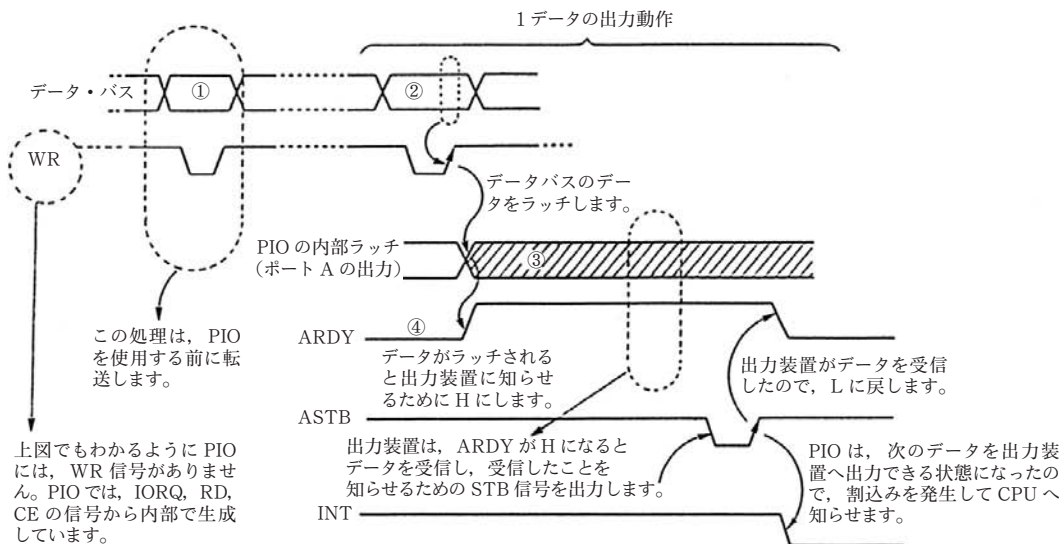
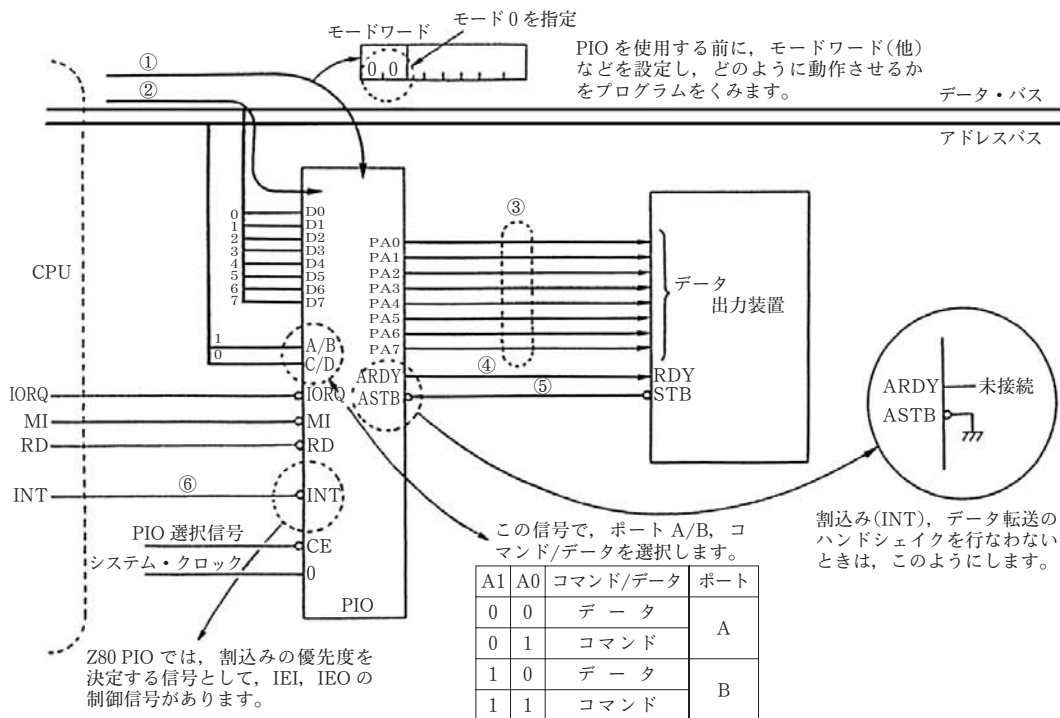
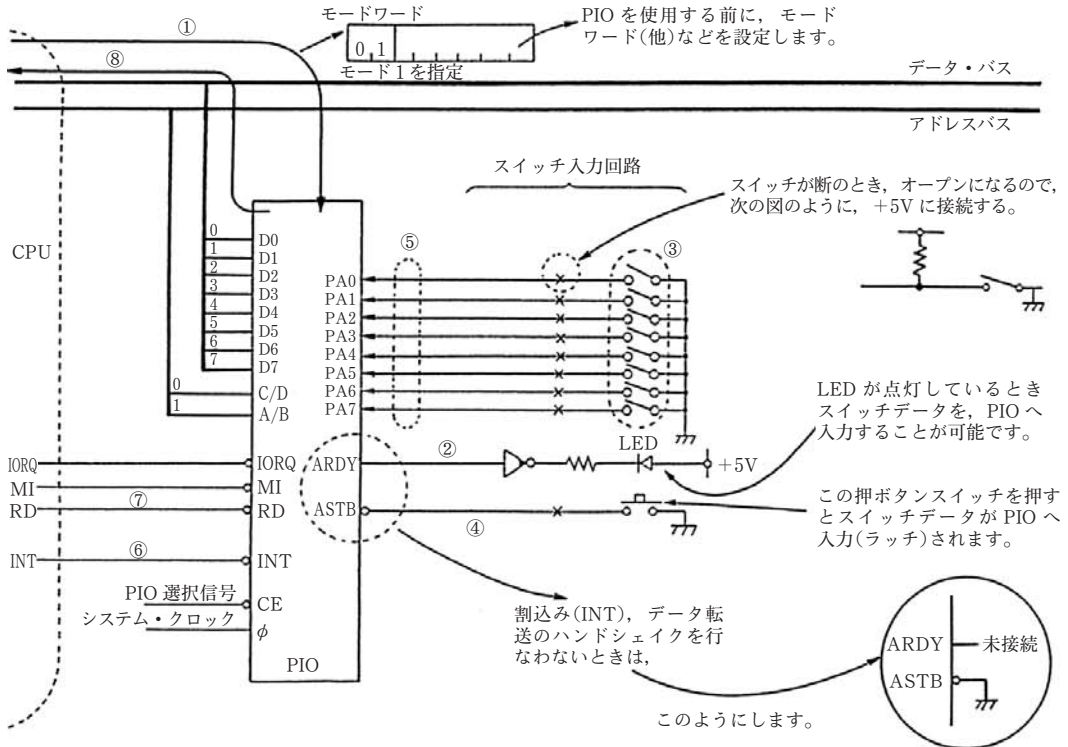


図9 PIO モード0の動作

(3) モード2の動作

このモードでは、ポート A のみ使用して、入力ポート、および出力ポートとして使用することができる。



この処理は、PIOを使用する前に行ないます。

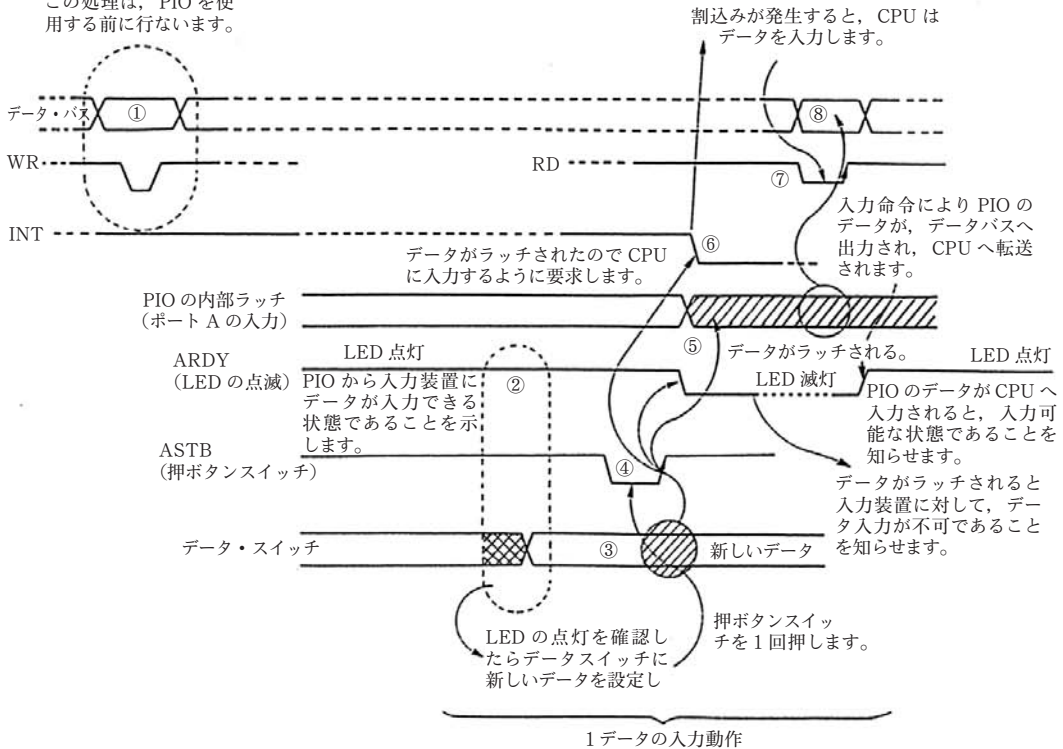


図10 PIOモード1の動作

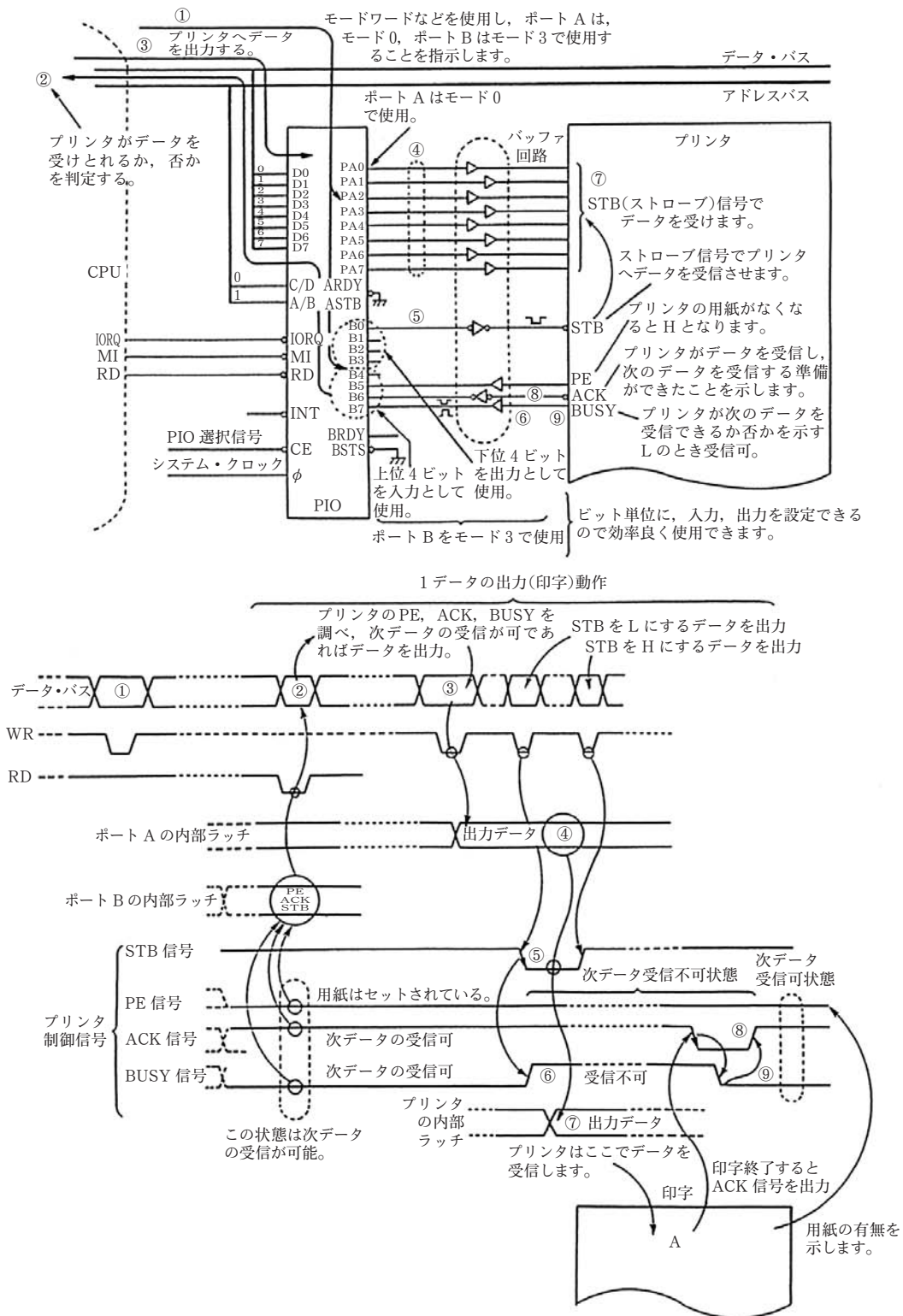


図 11 プリンタインタフェースの例と動作

信号はポート A の ARDY, ASTB 信号, およびにポート B の BRDY, BSTB 信号を使用し, ハンドシェイク制御を行う動作が可能である。

(4) モード 3 の動作

モード 0~2 の動作では, ポート単位に入力, 出力として使用したが, このモードでは, 各ビットに入力, 出力を決めて使用できる。このモードが, Z80 PIO の大きな特徴の一つである。

このモード 3 および, (1)でのモード 0 を利用した例として, プリンタインタフェースへの応用を図 11 に示す。

5.2 シリアルインタフェースの動作

Z80 SIO (Serial Input/Output Controller : SIO) はシリアル/パラレルのデータ変換を行う機能も持っている。

この Z80 SIO は, シリアル入出力ポートを 2 チャンネルもっており単純な調歩同期モードから, 高度な IBM-SDLC モードまでのいろいろなシリアル伝送形式に対応できるように設計されている。

SIO の機能を図 12 に示す。

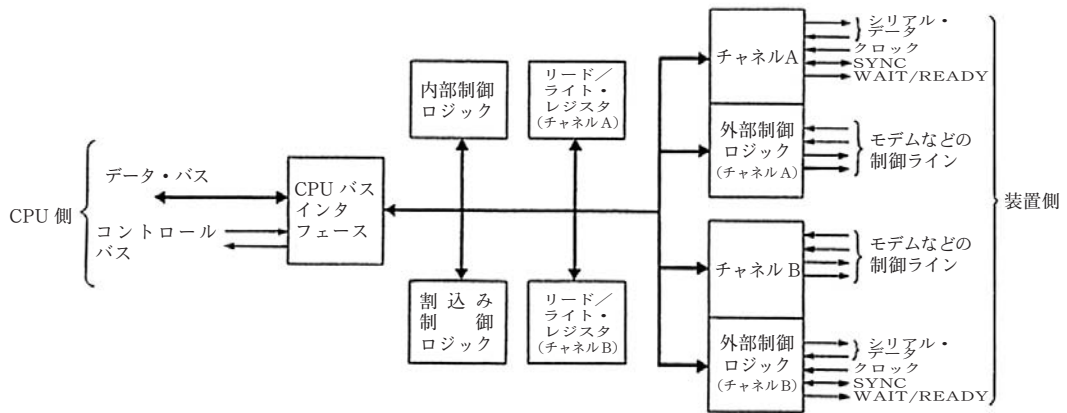


図 12 SIO の機能

引用・参考文献

渋井二三男・森田 博「マイコン・トレーニング初歩から応用まで」日刊工業新聞社

(Received Feb. 17, 2007)