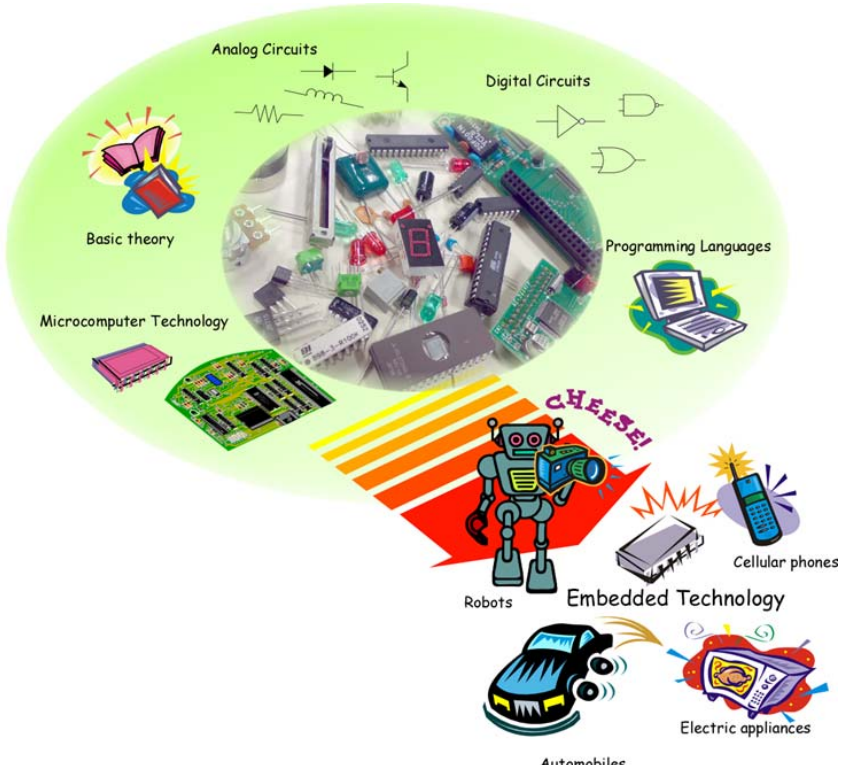
マイコン・ボード製作 （新ベーシック4回目、7月7日）

1. 組み込みシステムとマイコン
   - マイコンの構成
   - マイコンの選択ポイント
2. 実験ボード作成
   - 実験ボード概要(回路図)
   - ATmega644P
   - 周辺装置との接続
   - 部品リスト
   - 開発環境の紹介
   - はんだとはんだごての使い方
   - 動作確認

奈良高専
電気工学科
土井　滋貴



デジタル回路からマイコンへ

- 前回のデジタル回路では各種デジタルIC（AND,OR,Flip Flopなど）を組み合わせ、相互に接続することによって機能を確認した.
- 機能を変更するには使用するICやIC間の接続を変更しなければならない.

さまざまなデジタル回路（特に演算機能）を集積化

プログラムによって機能を変更可能

マイコン
(Micro Computer, Micro-Processor, Micro-Controller)

コンピュータの歴史

- コンピュータ（電子計算機）
  - 1942年　ABC（Atanasoff-Berry Computer）：　アイオア州立大
  - 1946年　ENIAC (Electronic Numerical　Integrator and Computer)　：ペンシルバニア大
    - 用途：　暗号解読，弾道計算など
    - 真空管18,000本，総重量30トン，消費電力150kW
    - 計算能力　加算：5,000回/秒，乗算（10桁）：14回/秒
  - 1949年　EDSAC（Electronic Delay Storage Automatic Calculator）
    - 世界初の実用的なプログラム記憶式電子計算機
  - 1950年代　商用コンピュータの登場（UNIVAC I，IBM 701）

真空管からトランジスタ・集積回路へ
磁気コアメモリから半導体メモリへ

- 大型コンピュータ（メインフレーム：IBM社 System/360など）
- ミニコンピュータ／オフィスコンピュータ・・・・コンピュータの小型化
- マイコン（マイクロプロセッサ／マイクロコンピュータ）
  - 1971年　i4004 ： Intel社
    - 電卓の使用が急増
      ⇒　機能変更が可能な算術計算用LSIとして開発（4ビットMPU）

コンピュータの構成

- 汎用コンピュータの五大機能
  演算，記憶，制御，入力，出力

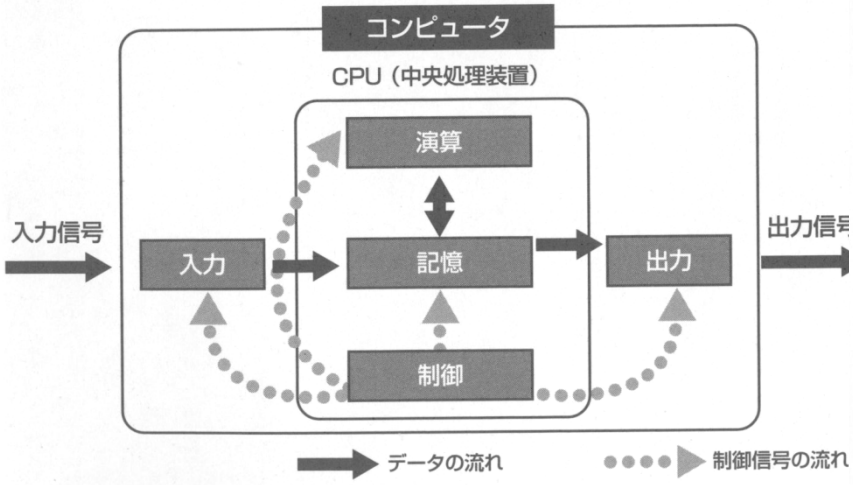


図 3-1-1　汎用コンピュータの構成（五大機能）

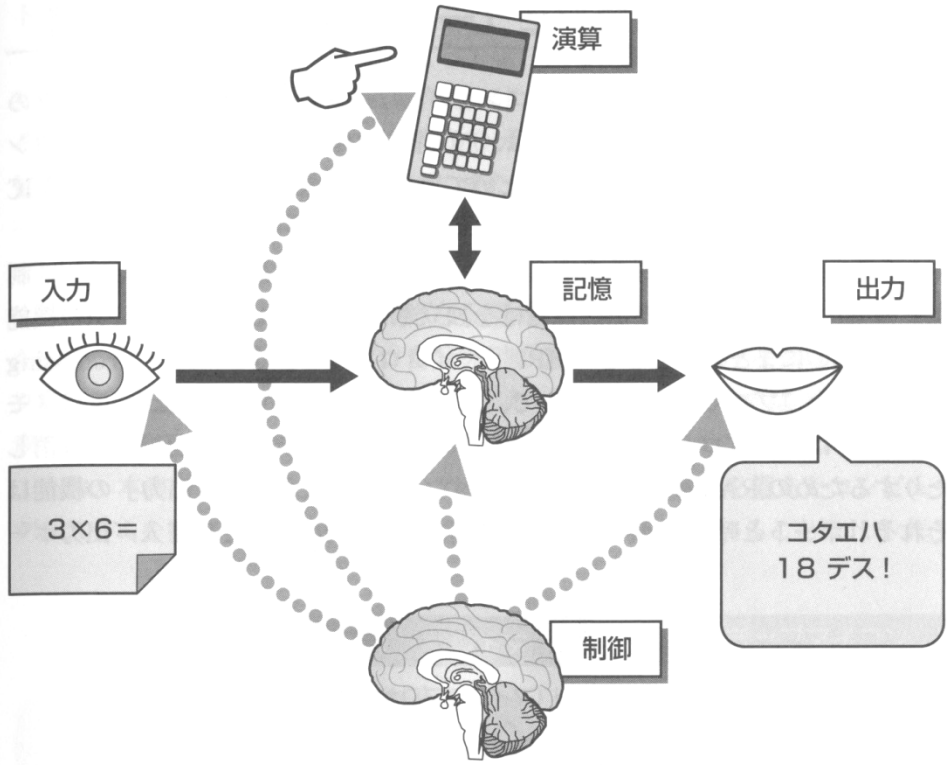
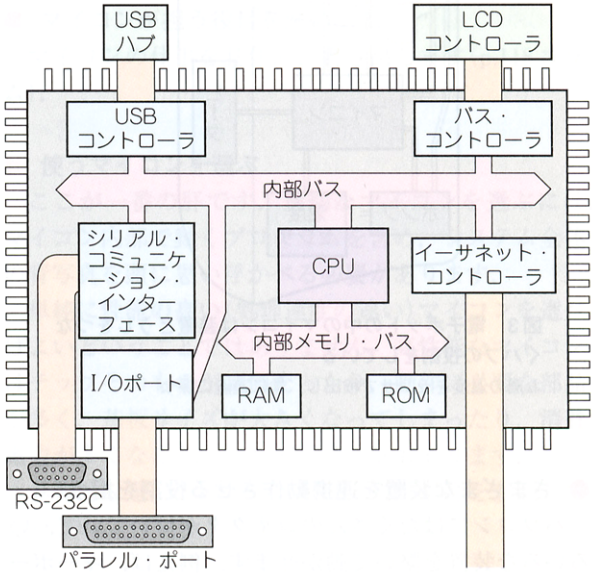


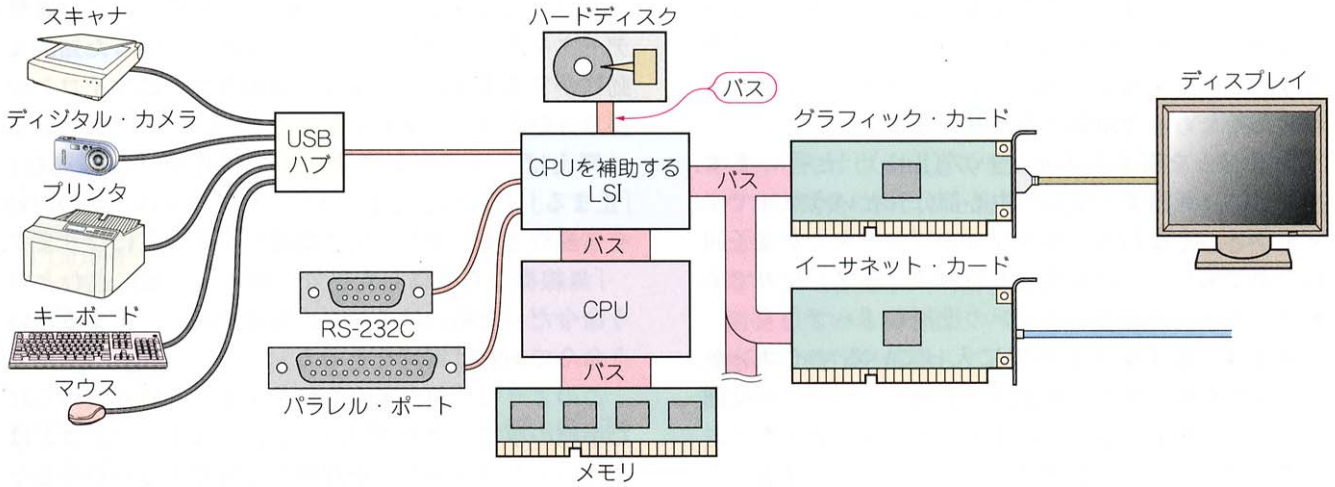
図 3-1-2　五大機能を人間にたとえると

マイコンの分類

- マルチチップ・マイコン
  - ⇒ CPU,メモリ(ROM,RAM),周辺IC など複数のIC(チップ)で構成
- シングルチップ・マイコン (ワンチップマイコン)
  - ⇒ 1つのICに集積化
  - ⇒ 組み込みシステムでは主にこちら



シングルチップ・マイコンのチップ内部構成

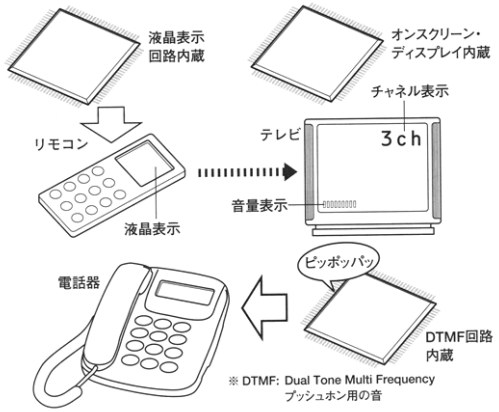


パソコン(マルチチップ・マイコン)の基本構成

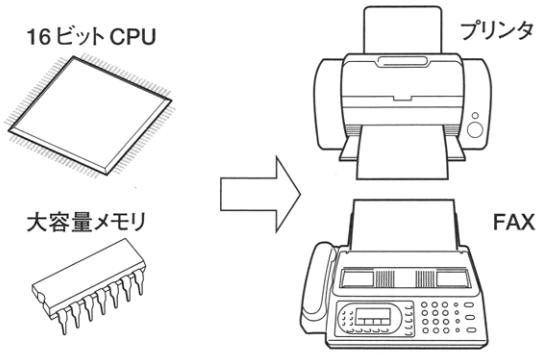
引用元:トランジスタ技術, 2007年5月号, CQ出版社, p.115

処理能力による分類

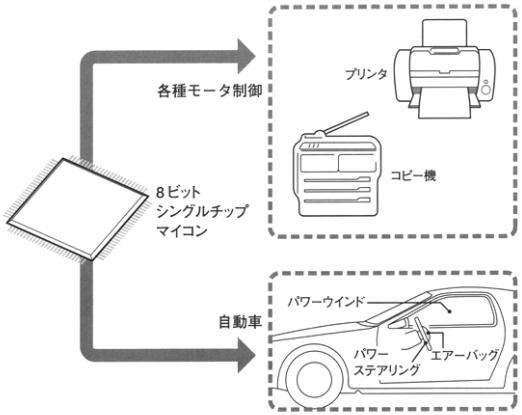
- 4ビットマイコン,8ビットマイコン,16ビットマイコン,32ビットマイコン
  ⇒ 1つの命令で扱うことのできるビット数が大きいほど高性能



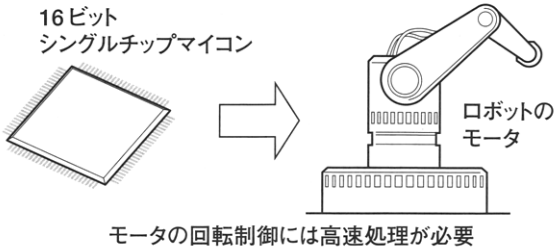
4ビットマイコンの応用例



画像を扱うには大容量メモリが必要

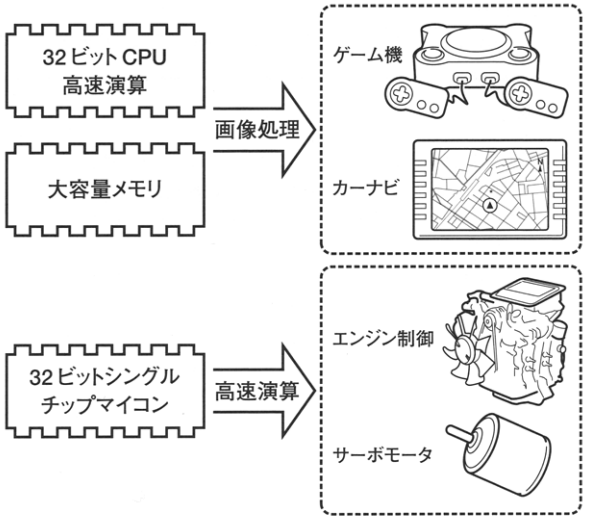


8ビットマイコンの応用例



モータの回転制御には高速処理が必要

16ビットマイコンの応用例



32ビットマイコンの応用例

引用元:「マイコン入門講座」,大須賀威彦 著, 電波新聞社, p.25~29

コンピュータの基本構成

- CPU（MPU）　：　命令語の実行
- メモリ　：　命令語・データの記憶
- 入出力部　：　外部デバイスとのデータのやり取り

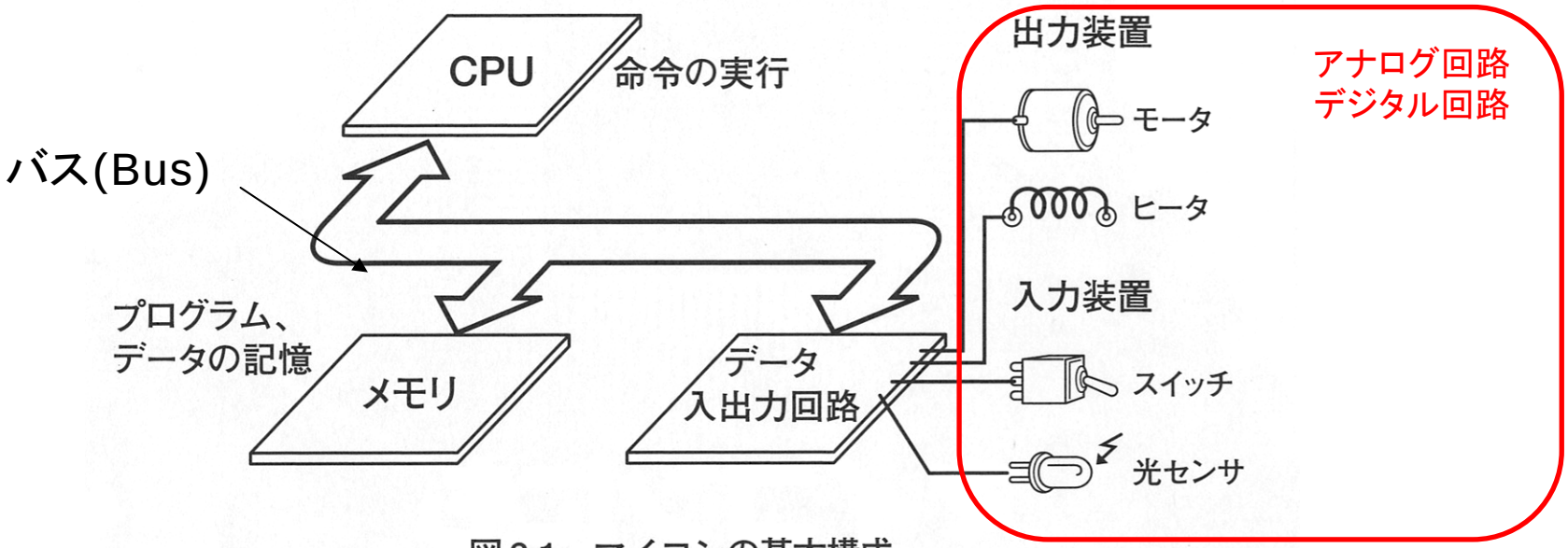


図 2.1　マイコンの基本構成

CPU: Central Processing Unit
MPU: Micro-Processing Unit

引用元:「マイコン入門講座」,大須賀威彦 著, 電波新聞社, p.20

バス（Bus：母線）

- MPU(CPU)⇔メモリ⇔入出力部でのデータ転送
  - アドレスバス(Address Bus)
    - MPU　⇒　メモリなど（片方向）
    - データの読み出し・書き込み場所を指定するための信号線
  - データバス(Data Bus)
    - MPU ⇔　メモリなど（双方向）
    - 実際のデータが行き来するための信号線
  - コントロールバス(Control Bus)
    - MPU,メモリ,入出力部でのデータの転送を行なうためのタイミングや指示（命令）、状態を伝えるための信号線

バスによる構成要素間の接続

- バスへの構成要素の接続により機能を追加

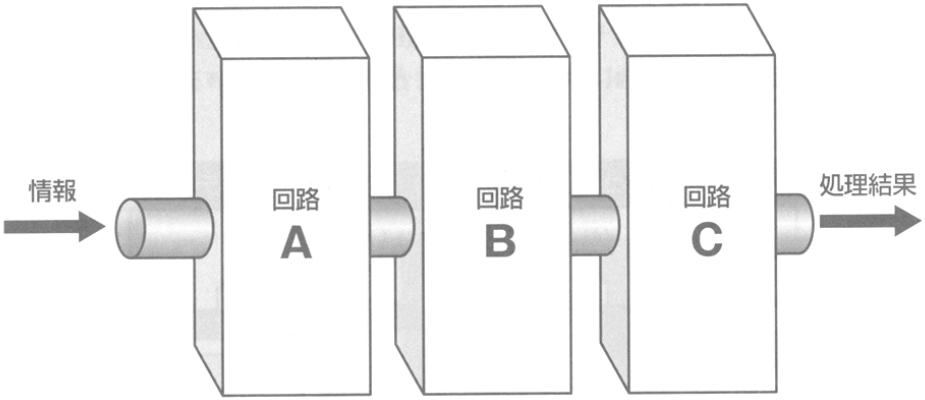


図 3-3-1　一般的な電子回路の接続法

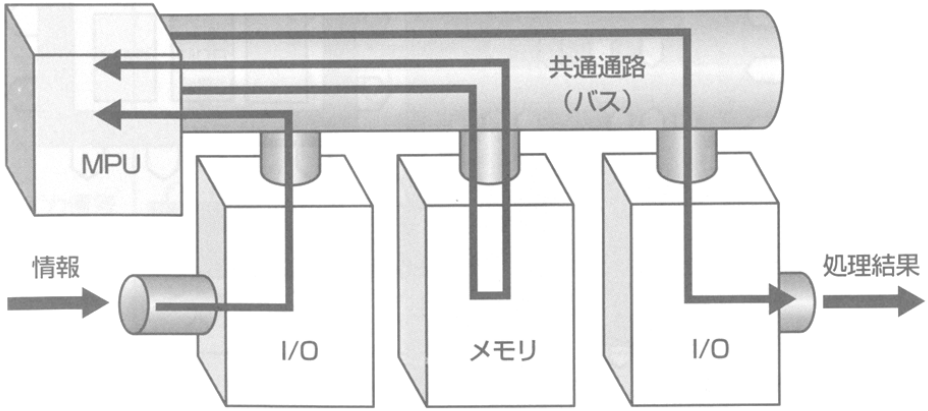


図 3-3-2　バス構成によるマイコン部の接続法

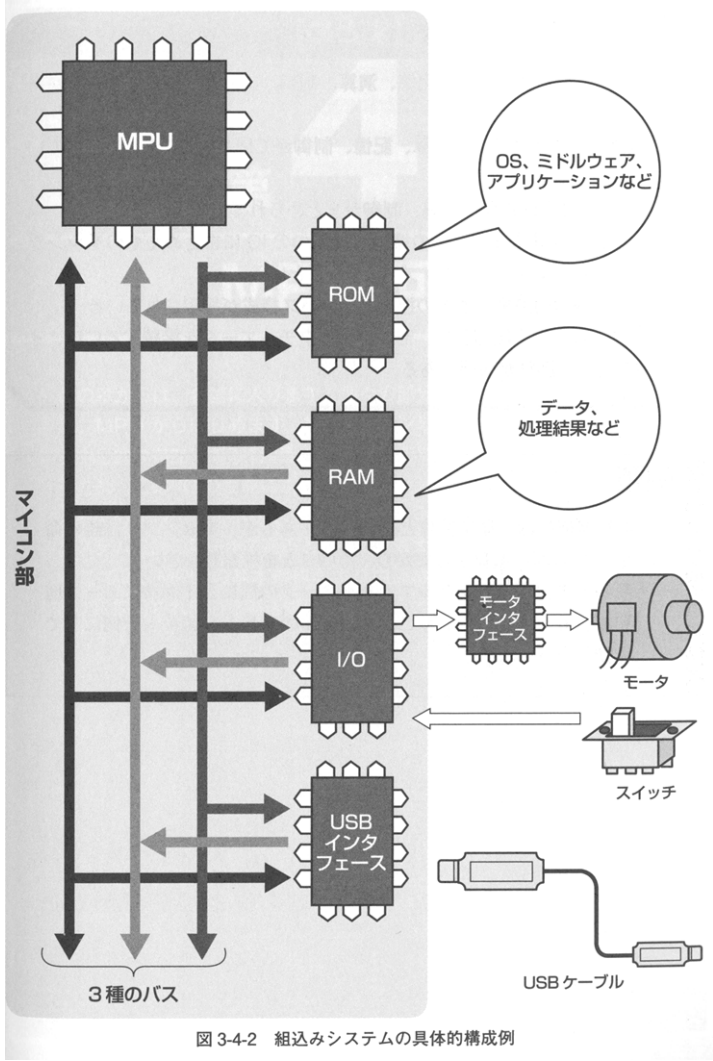
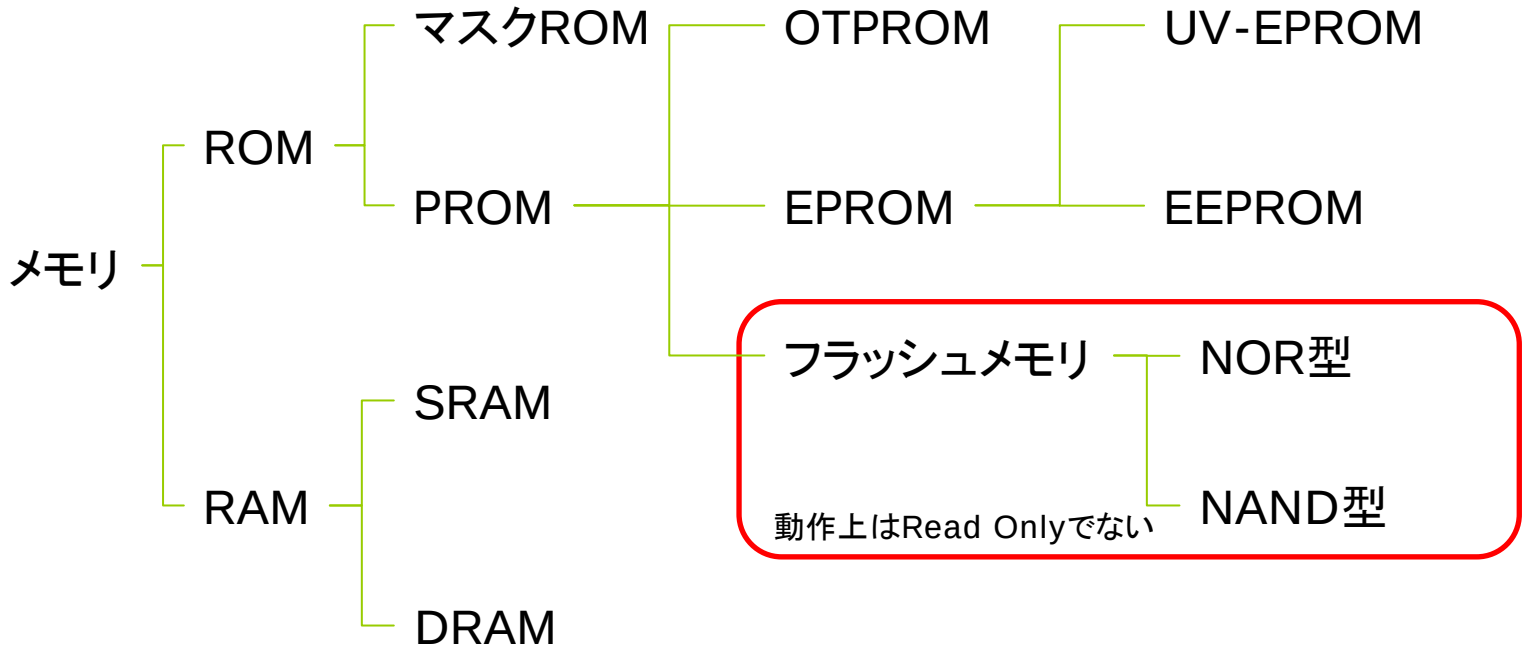


図 3-4-2　組込みシステムの具体的構成例

メモリ

- ROM（Read Only Memory）
  - 電源が消えても書き込まれた情報（データ）が消えない
  - 基本的には情報（データ）の書き換えができない
- RAM（Random Access Memory）
  - 情報（データ）を自由に書き換え可能であるが、電源を切ると内容が消失

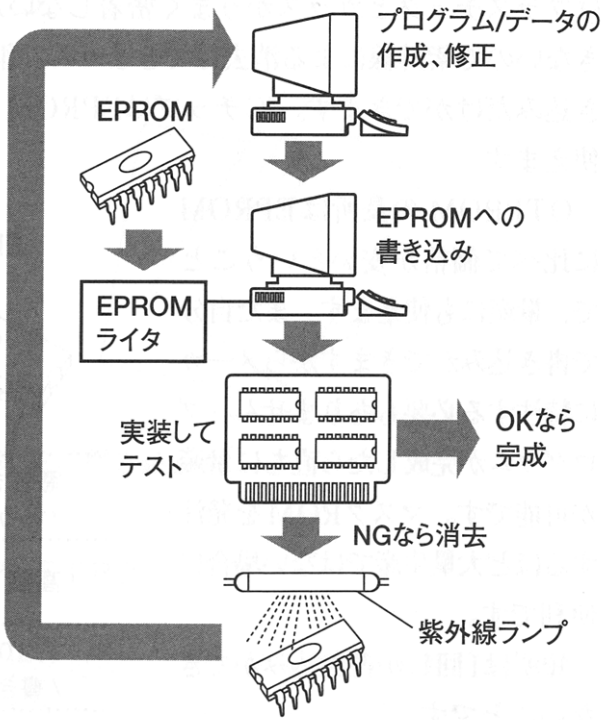


メモリの種類

ROM

ROMの分類

- マスクROM:製造時にデータを作り込む
- PROM(Programmable ROM)
  - OTPROM(One Time Programmable ROM) 消去不可
  - EPROM (Erasable Programmable ROM) 製造後に書き込み／消去が可能
    - UV-PROM (Ultra Violet Programmable ROM): 紫外線によって情報(データ)の消去を行なうEPROM
    - EEPROM (Electrically Erasable Programmable ROM): 電気的に情報(データ)の消去・書き込みが可能
  - フラッシュメモリ(Flash Memory)
    - NOR型:高速アクセス(書き込みは遅い)
    - NAND型:回路規模小⇒大容量化



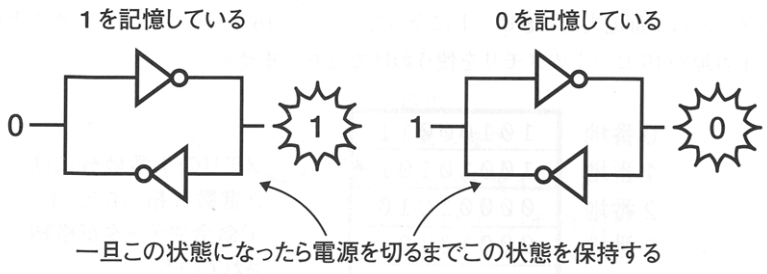
UV-PROM

引用元:「マイコン入門講座」,大須賀威彦 著,
電波新聞社, p.45

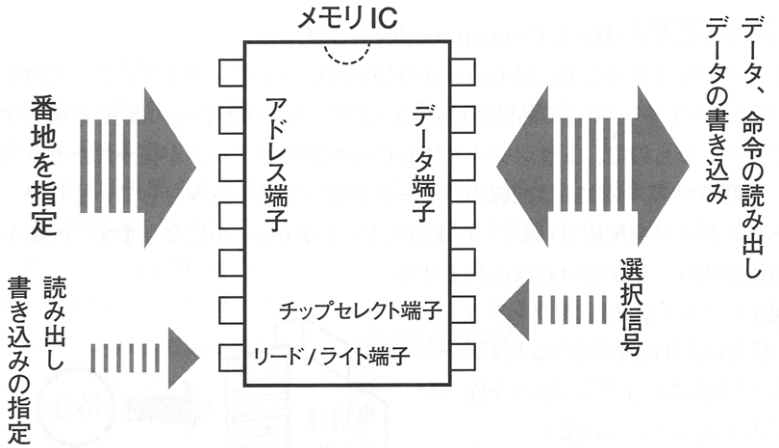
SRAM

- SRAM（Static RAM）
  - フリップ・フロップ（Flip Flop）に1ビットの情報を保持
  - 1ビット当たり4～6個のトランジスタ
  - 低消費電力（保持だけなら数μW）で高速（アクセス時間：10ns以下のものもある）であるが比較的高価

回路が複雑になり集積度を上げにくい



SRAMの原理



SRAMチップの端子

引用元：「マイコン入門講座」,大須賀威彦 著，電波新聞社，p.42,43

メモリチップの例（SRAM）

データ・シート

NEC

MOS 集積回路
MOS Integrated Circuit

μPD43256B

256K ビット CMOS スタティック RAM
32K ワード×8 ビット

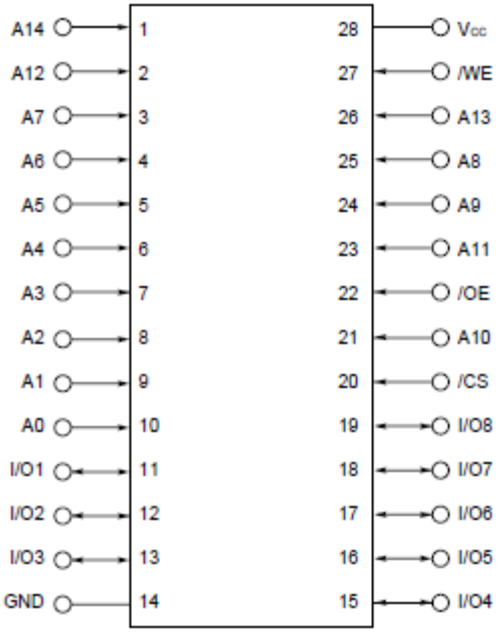
μPD43256B は 262,144 ビット（32,768 ワード×8 ビット）の CMOS スタティック RAM です。低消費電力タイプなので，バッテリ・バックアップに最適です。さらに A，B バージョンは低電圧動作ができます。

外形は，28 ピン・プラスチック DIP，28 ピン・プラスチック SOP，28 ピン・プラスチック TSOP（I）（8 × 13.4 mm）です。

特　徴

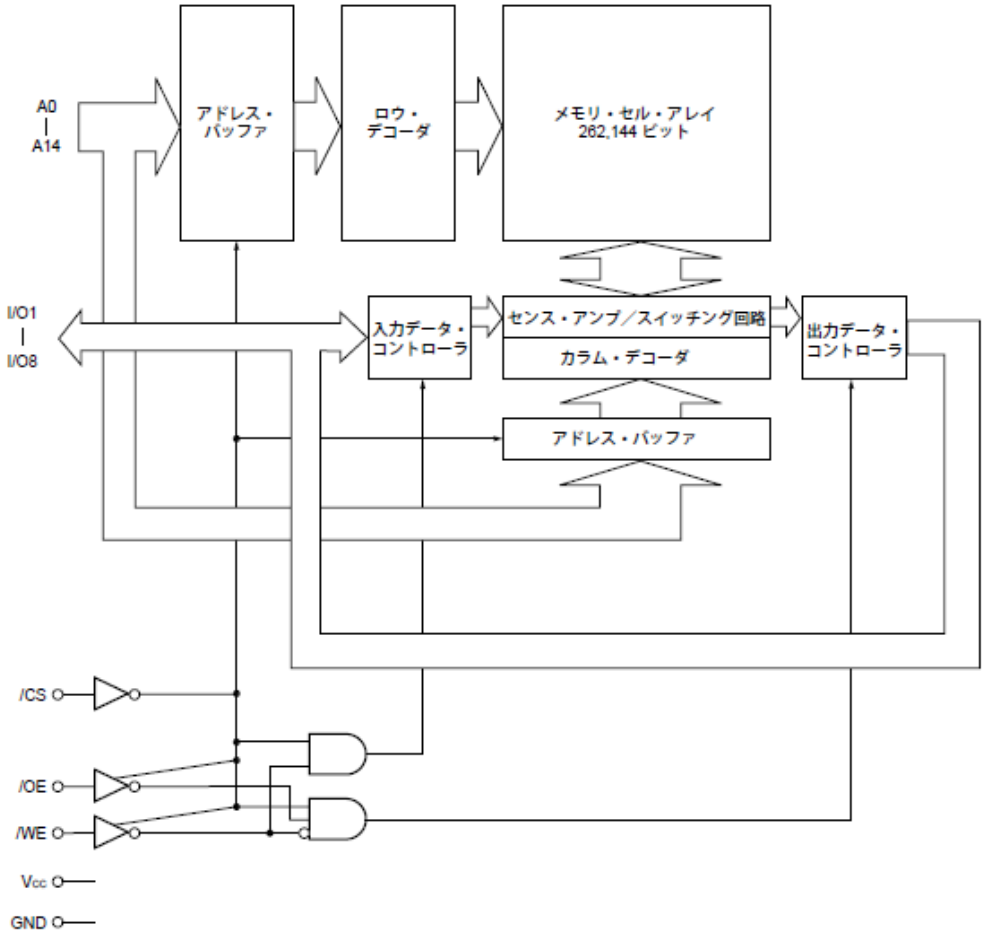
- ワード構成：32,768 ワード×8 ビット
- 高速アクセス時間：70, 85, 100, 120 ns（MAX.）
- 低電圧動作（A バージョン：Vcc = 3.0～5.5 V，B バージョン：Vcc = 2.7～5.5 V）
- 低電源電圧データ保持：2.0 V（MIN.）
- アウトプット・バッファを制御する/OE 端子を備えています。

μPD43256BCZ70LL



A0～A14	：アドレス入力
I/O1～I/O8	：データ入出力
/CS	：チップ・セレクト入力
/WE	：ライト・イネーブル入力
/OE	：アウトプット・イネーブル入力
Vcc	：電源
GND	：グランド

ブロック図

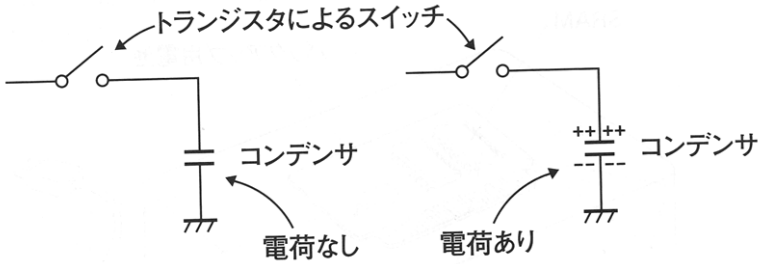


DRAM

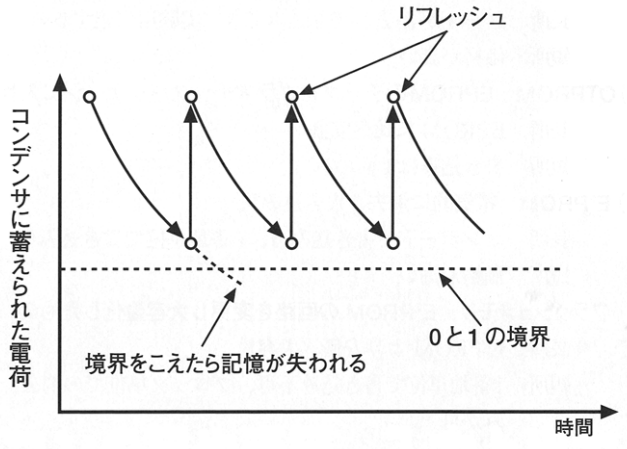
- DRAM（Dynamic RAM）
  - 1ビット当たり1つのトランジスタと1つのコンデンサで構成
  - コンデンサに電荷が蓄えられているかどうかで”1”か”0”を記憶
  - SRAMより大容量・安価
  - コンデンサの電荷は時間とともに減少（放電）

    ⇒ リフレッシュ（情報が消失する前にデータを書き直す：約15μs毎）
  - DRAMの種類
    - SDRAM (Synchronous DRAM) ： メモリバスクロックに同期してデータ転送を高速化
    - DDR SDRAM (Double Data Rate SDRAM) ：メモリバスクロックの2倍でデータ転送

回路が単純で集積度が上げやすい



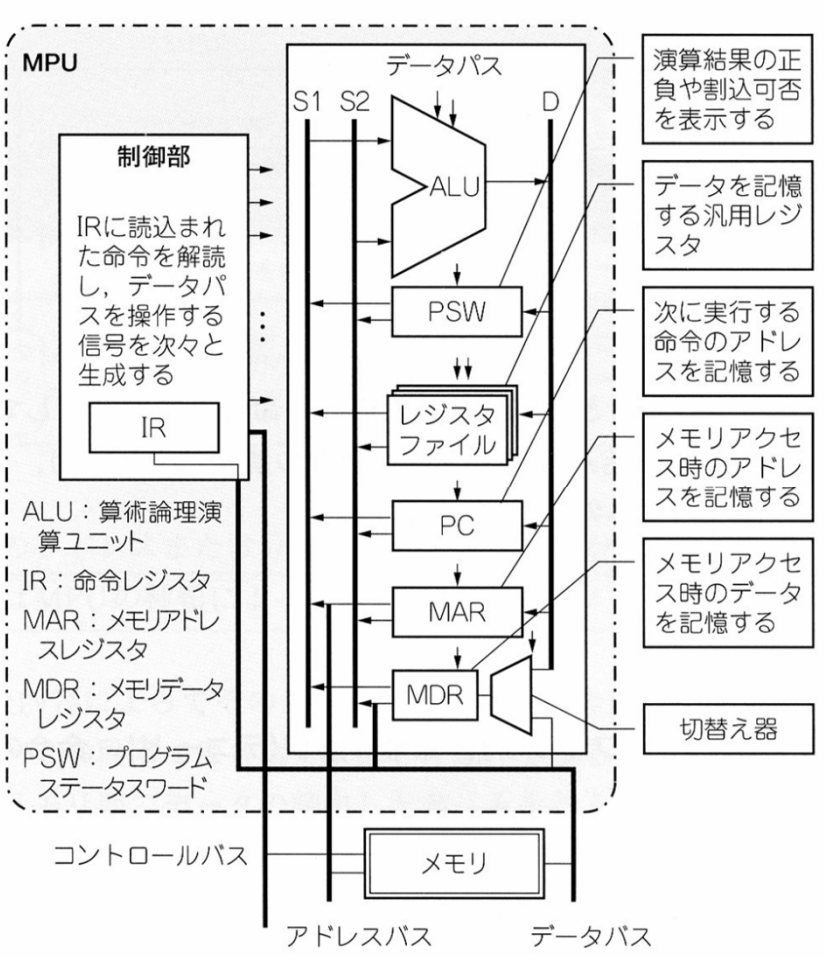
DRAMの原理



リフレッシュ

引用元：「マイコン入門講座」,大須賀威彦 著，電波新聞社，p.50,51

CPU（MPU）の内部構成



引用元:「組み込みソフトウェア技術者試験クラス2対策ガイド」,
(社)組込みシステム技術協会 編・著, CQ出版社, p.13

制御

- 命令レジスタ(Instruction Register: IR)
  命令レジスタに読み込まれた命令語を解読し、データパスを操作

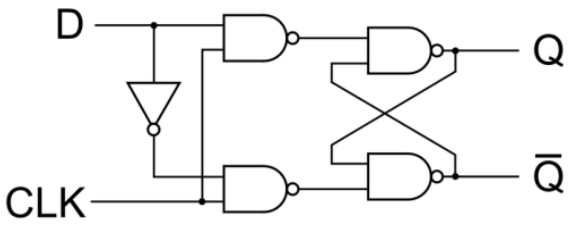
データパス

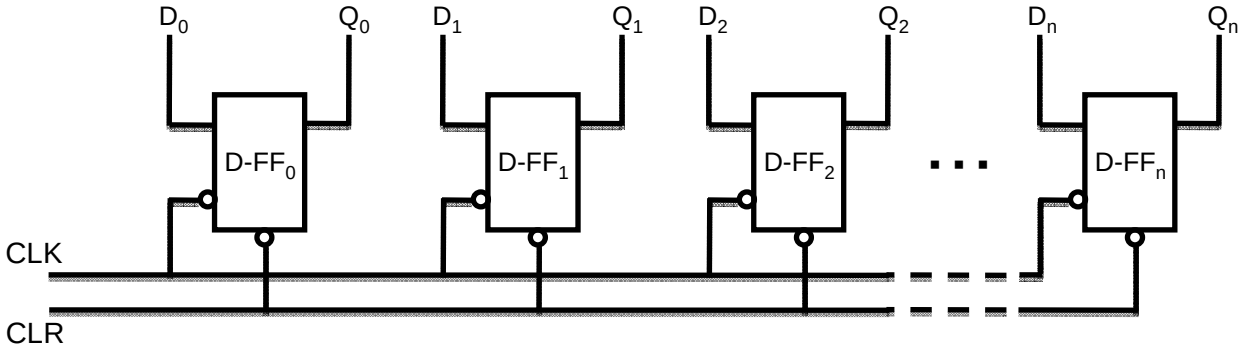
- プログラムカウンタ(Program Counter: PC)
  次に読み込む命令語が格納されたアドレスを指定
- レジスタファイル
  データを記憶する汎用レジスタで演算結果などを一時的に記憶
- 算術論理演算ユニット
  (Arithmetic and Logical operation Unit: ALU)
  算術演算・論理演算を実行
- メモリアドレスレジスタ
  (Memory Address Register: MAR)
  メモリへアクセスする際のアドレスを一時的に記憶
- メモリデータレジスタ(Memory Data Register: MDR)
  メモリとのデータ転送を行なう際にデータを一時的に記憶
- プログラムステータスワード
  (Program Status Word: PSW)
  演算結果が正か負か、オーバフローが発生したか、割り込みが可能かどうかなどの情報

CPUコア

レジスタ：CPUコア内に配置されたメモリの一種。
CPUの動作状態や行動を決定するパラメータを格納する専用レジスタとデータの保持等に使われる汎用レジスタに大きく分かれる。

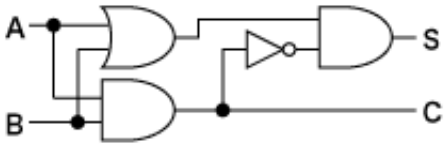
専用レジスタ：プログラムカウンタ、ステータスレジスタ
コントロールレジスタ、アドレスレジスタ
スタックポインタ、データレジスタ
インストラクションレジスタなど

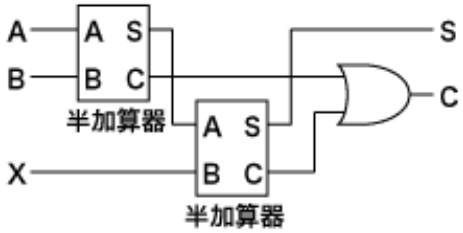




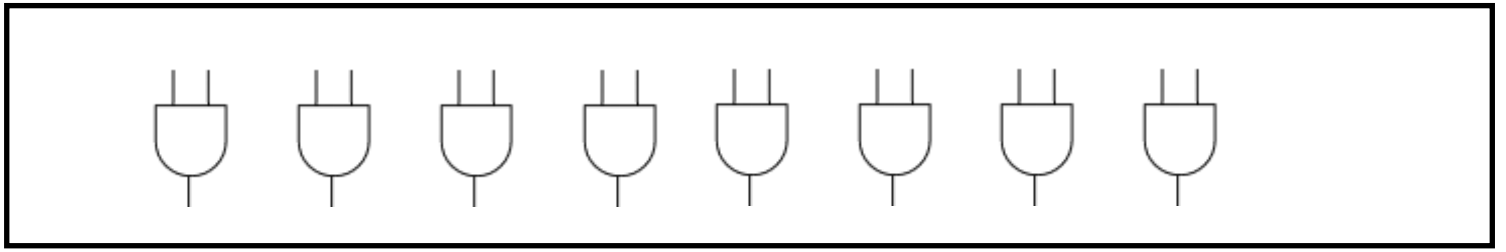
CPUコア

演算装置(ALU)：論理演算や四則演算などの算術演算に特化したユニット。論理回路で構成された演算処理回路、一時記憶レジスタ(ラッチ)、入出力用の配線で構成される。

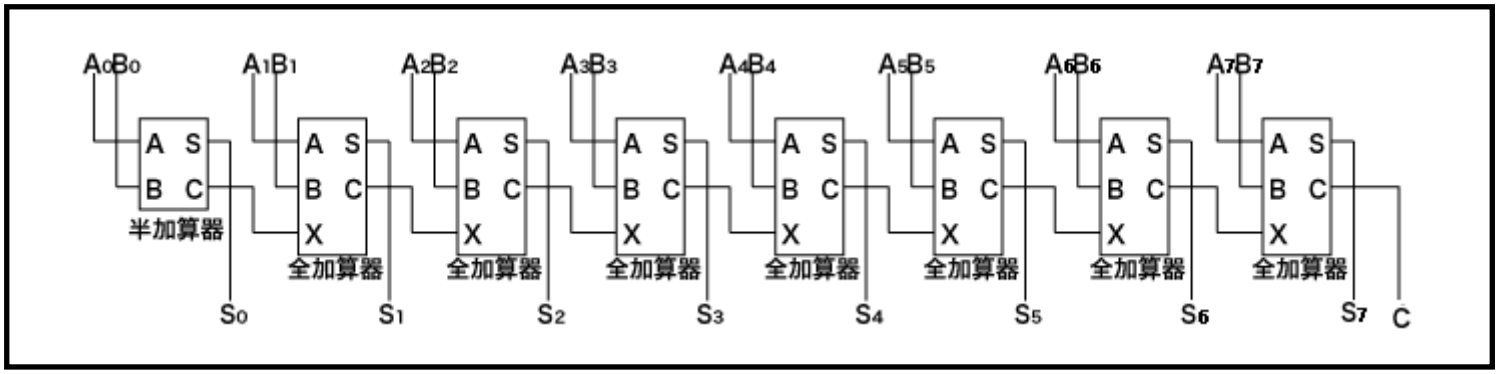




AND



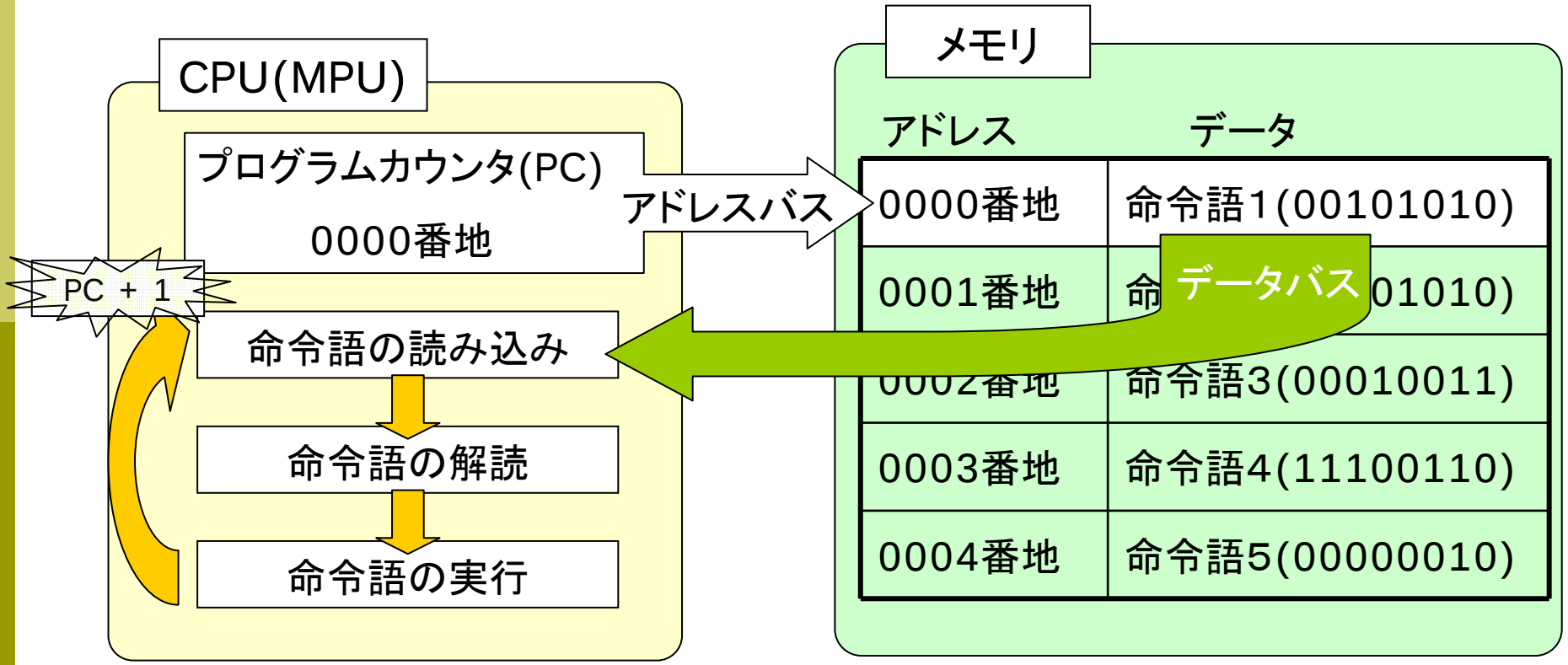
ADD



CPUの動作 プログラム記憶式（ストアードプログラム方式，ノイマン方式）

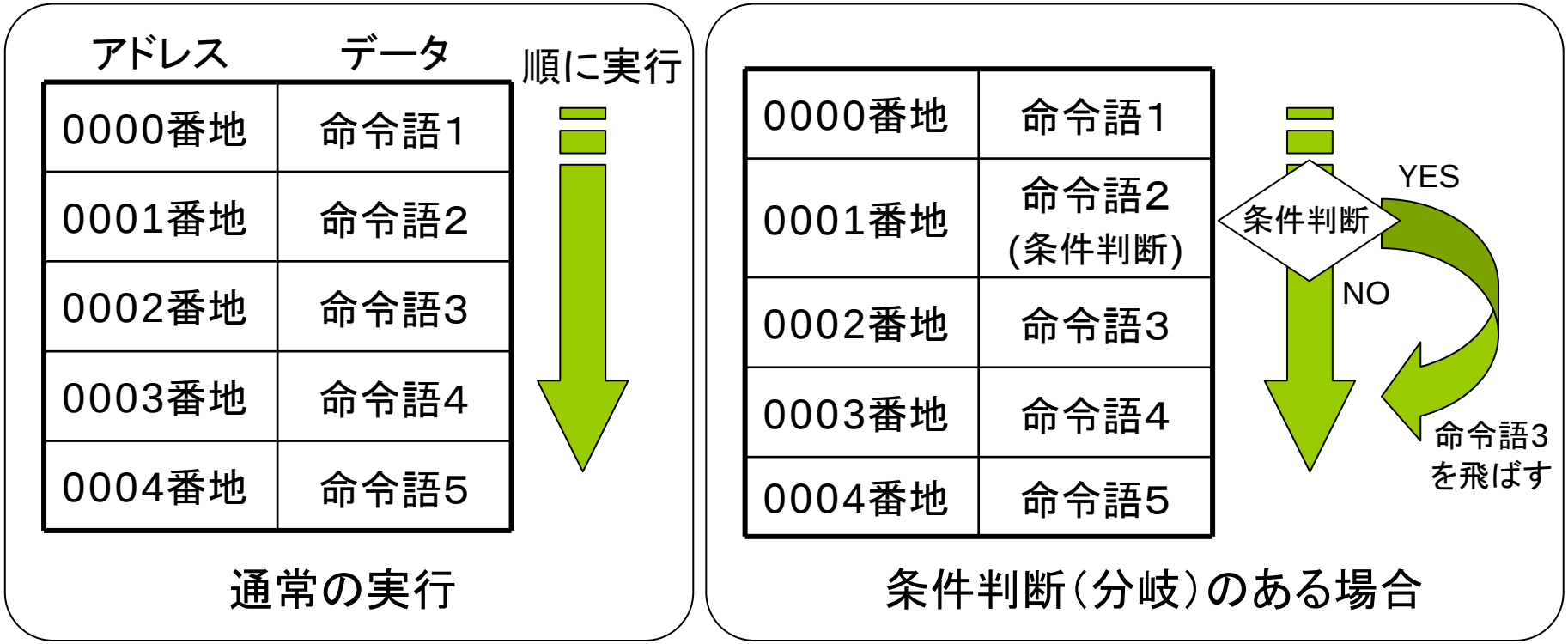
- メモリに記憶されている命令を1つずつ順に実行
  - 命令語の読み込み（Fetch）
  - 命令語の解読（Decode）
  - 命令語の実行（Execute）

繰り返し



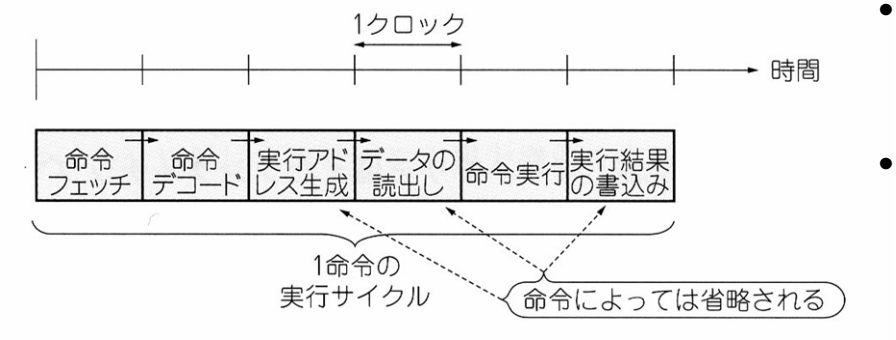
命令語の実行順序

- 命令語の実行順序はプログラムカウンタ(PC)に従う
  - ⇒ PCは次に読み込む命令語が格納されているアドレスを指定
  - ⇒ 命令語が読み込まれるとPCは1増加
  - ⇒ ただし、命令語の中には強制的にPCを書き換えるものが存在



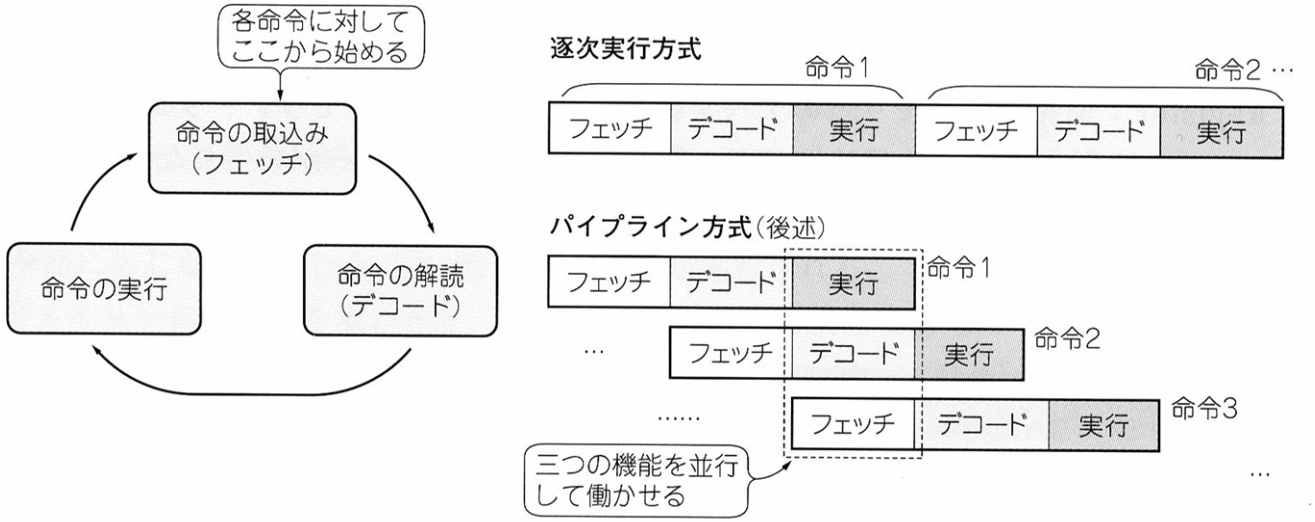
命令語の処理方式

命令語の実行サイクル



- クロック: CPUが動作するタイミングをとるための周期的な基本信号
  ⇒ クロック周波数[Hz]
- CPI (Clock Cycle Per Instruction): 各命令語の実行に要するクロック数

逐次処理方式とパイプライン方式



引用元:「組み込みソフトウェア技術者試験クラス2対策ガイド」, (社)組込みシステム技術協会 編・著, CQ出版社, p.14,15

パイプライン処理方式

- 特徴
  - 1つの命令語の実行が終わるのを待たずに次の命令語を読み込み、並列的に処理を実行
  - 理想はCPI（Clock Cycle Per Instruction）＝ 1
- パイプラインハザード

  ⇒　並列処理が行なえなくなる状態　⇒　CPIが増加
  - 構造ハザード
    - それぞれのステージ（Fetch, Decode, Executeなど）を実行するハードウェア資源が競合し同時に処理できなくなること
  - データハザード
    - 次の命令語が先行する処理結果を用いるために並列処理が実行できなくなること（先行する処理結果が出るまで待ち状態）
  - 制御ハザード
    - 分岐が成立したために、分岐先の命令語を読み込み（Fetch）しなければならなくなること（すでに処理していた次の命令語を破棄し、別の命令語をFetch）

CPUの分類（CISCとRISC）

- コンピュータの処理速度向上のため‥
  - CISC (Complex Instruction Set Computer)

    ⇒ たくさんの命令語を用意し、プログラムのステップ数を削減
    - CPU(MPU)内部の構成が複雑
    - 命令語の解読（Decode）に時間がかかる
    - 命令の処理が複雑になれば実行に要するクロック数（CPI）が多くなる（さまざまなCPIの命令語が存在） ⇒ パイプライン処理方式に不適
  - RISC (Reduced Instruction Set Computer)

    ⇒ 命令の種類を減らし、かつ単純な命令とすることにより処理効率を向上
    - 命令語が単純であるのでCPU(MPU)内部の構成が簡素
    - 命令語の実行に要するクロック数(CPI)が少ない ⇒ 高速クロック周波数
    - 各命令語のCPIを統一することによりパイプライン処理方式に適合
    - まとまった動作をさせるには多くの命令が必要
  - その他
    - ベクトルプロセッサ：配列演算の同時並列実行に着目
    - スーパスカラ方式：パイプラインを複数用意し、同時実行させる

    など

命令語

- 機械語命令（マシン語命令）
  - CPU(MPU)が理解するための命令語
  - “0”と”1”の組み合わせで表現
  - 人には理解できない（対応付けができない,覚えられない）
- アセンブリ言語(Assembly Language)
  - 機械語命令と一対一に対応
  - ニーモニック(Mnemonic)と呼ばれるアルファベットで命令を表現（意味の分かる単語・略語を使用）
- アセンブラ(Assembler)
  - アセンブリ言語を機械語命令に翻訳するソフトウェア

図3.7　アセンブリ言語と機械語

引用元:「マイコン入門講座」,大須賀威彦 著, 電波新聞社, p.37

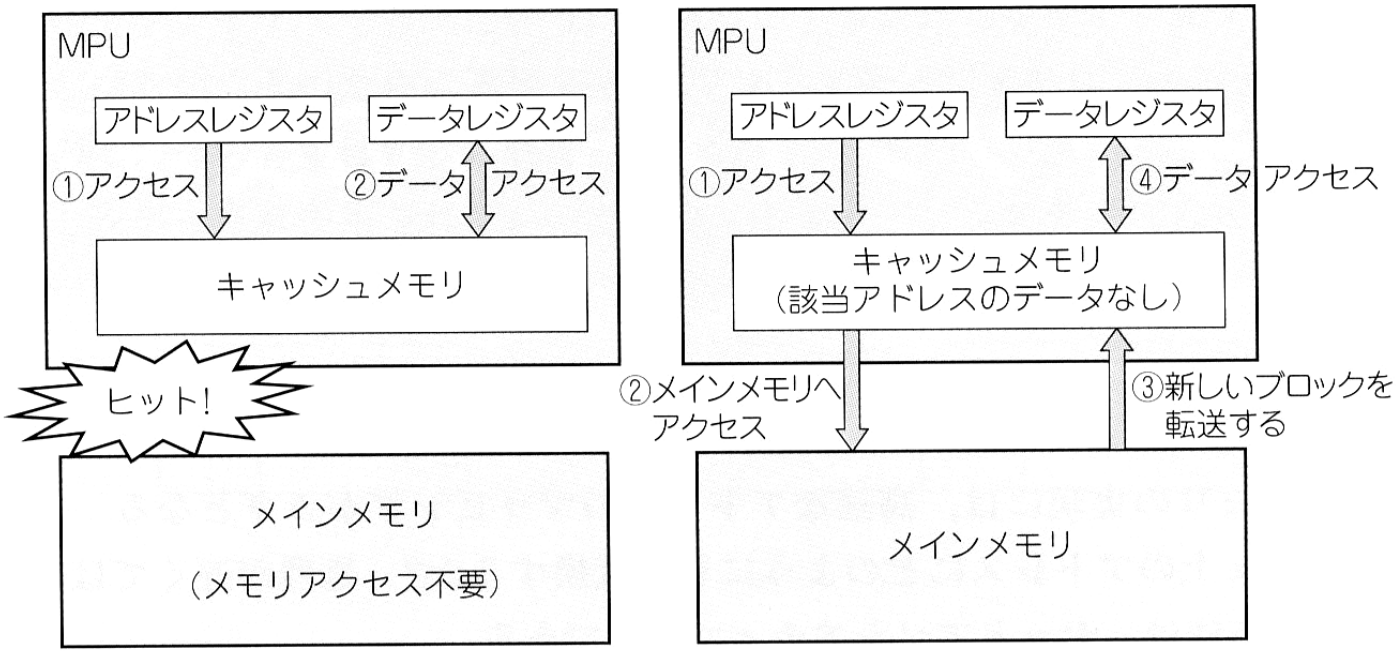
命令語一覧(例:AVRマイコンの場合)

- アセンブリ言語

ニーモニック	オペランド	意味	動作	フラグ	クロック
算術、論理演算命令					
ADD	Rd,Rr	汎用レジスタ間の加算	Rd ← Rd + Rr	I,T,H,S,V,N,Z,C	1
ADC	Rd,Rr	キャリーを含めた汎用レジスタ間の加算	Rd ← Rd + Rr + C	I,T,H,S,V,N,Z,C	1
ADIW	Rd,K6	即値の語(ワード)長加算	RdH:RdL ← RdH:RdL + K6	I,T,H,S,V,N,Z,C	2
SUB	Rd,Rr	汎用レジスタ間の減算	Rd ← Rd − Rr	I,T,H,S,V,N,Z,C	1
SUBI	Rd,K	汎用レジスタから即値の減算	Rd ← Rd − K	I,T,H,S,V,N,Z,C	1
SBIW	Rd,K6	即値の語(ワード)長減算	RdH:RdL ← RdH:RdL − K6	I,T,H,S,V,N,Z,C	2
SBC	Rd,Rr	キャリーを含めた汎用レジスタ間の減算	Rd ← Rd − Rr − C	I,T,H,S,V,N,Z,C	1
SBCI	Rd,K	汎用レジスタからキャリーと即値の減算	Rd ← Rd − K − C	I,T,H,S,V,N,Z,C	1
AND	Rd,Rr	汎用レジスタ間の論理積(AND)	Rd ← Rd AND Rr	I,T,H,S,0,N,Z,C	1
ANDI	Rd,K	汎用レジスタと即値の論理積(AND)	Rd ← Rd AND K	I,T,H,S,0,N,Z,C	1
OR	Rd,Rr	汎用レジスタ間の論理和(OR)	Rd ← Rd OR Rr	I,T,H,S,0,N,Z,C	1
ORI	Rd,K	汎用レジスタと即値の論理和(OR)	Rd ← Rd OR K	I,T,H,S,0,N,Z,C	1
EOR	Rd,Rr	汎用レジスタ間の排他的論理和(Ex-OR)	Rd ← Rd EOR Rr	I,T,H,S,0,N,Z,C	1
COM	Rd	1の補数(論理反転)	Rd ← $FF − Rd	I,T,H,S,0,N,Z,1	1
NEG	Rd	2の補数	Rd ← $00 − Rd	I,T,H,S,V,N,Z,C	1
SBR	Rd,K	汎用レジスタの(複数)ビット セット(1)	Rd ← Rd OR K	I,T,H,S,0,N,Z,C	1
CBR	Rd,K	汎用レジスタの(複数)ビット クリア(0)	Rd ← Rd AND ($FF − K)	I,T,H,S,0,N,Z,C	1
INC	Rd	汎用レジスタの増加(+1)	Rd ← Rd + 1	I,T,H,S,V,N,Z,C	1
DEC	Rd	汎用レジスタの減少(−1)	Rd ← Rd − 1	I,T,H,S,V,N,Z,C	1
TST	Rd	汎用レジスタのゼロとマイナス検査	Rd ← Rd AND Rd	I,T,H,S,0,N,Z,C	1
CLR	Rd	汎用レジスタの全0設定(=$00)	Rd ← Rd EOR Rd	I,T,H,0,0,0,1,C	1
SER	Rd	汎用レジスタの全1設定(=$FF)	Rd ← $FF	I,T,H,S,V,N,Z,C	1
MUL	Rd,Rr	符号なし間の乗算	R1:R0 ← Rd×Rr (U×U)	I,T,H,S,V,N,Z,C	2
MULS	Rd,Rr	符号付き間の乗算	R1:R0 ← Rd×Rr (S×S)	I,T,H,S,V,N,Z,C	2
MULSU	Rd,Rr	符号付きと符号なしの乗算	R1:R0 ← Rd×Rr (S×U)	I,T,H,S,V,N,Z,C	2
FMUL	Rd,Rr	符号なし間の固定小数点乗算	R1:R0 ← (Rd×Rr)<<1 (U×U)	I,T,H,S,V,N,Z,C	2
FMULS	Rd,Rr	符号付き間の固定小数点乗算	R1:R0 ← (Rd×Rr)<<1 (S×S)	I,T,H,S,V,N,Z,C	2

キャッシュメモリ(Cache Memory)方式

- 命令の読み出しなどにおいてメモリ(メインメモリ)とのデータのやり取りに要する時間を削減したい
- MPU近くに小容量・高速動作のメモリを配置し、実行中のソフト資源をコピーして処理を実行
- キャッシュメモリに目的とするアドレスのデータや命令があれば(ヒット)、それを使って実行　⇒　メインメモリへのアクセスが不要



(a) キャッシュがヒットした場合　　(b) キャッシュがミスヒットの場合

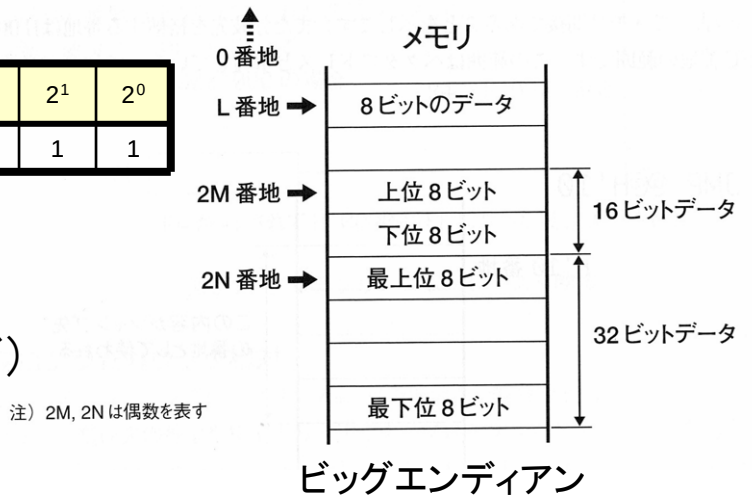
引用元:「組み込みソフトウェア技術者試験クラス2対策ガイド」, (社)組込みシステム技術協会 編・著, CQ出版社, p.29

数値の格納方式（エンディアン）

- 情報（数値・データ）はメモリに格納（記憶）される
- メモリのアドレスは8bit（1バイト：Bite）毎に割り振られる
- データサイズは様々　⇒　8bit毎に分割して格納
- 16bitデータの場合
  - 上位8bitを2M（偶数）番地，下位8bitを2M＋1（奇数）番地へ格納
    ⇒　ビッグエンディアン（H8マイコンなど）
  - 下位8bitを2M（偶数）番地，上位8bitを2M＋1（奇数）番地へ格納
    ⇒　リトルエンディアン（Intel x86，AVRなど）

上位8bit								下位8bit							
2^{15}	2^{14}	2^{13}	2^{12}	2^{11}	2^{10}	2^{9}	2^{8}	2^{7}	2^{6}	2^{5}	2^{4}	2^{3}	2^{2}	2^{1}	2^{0}
1	1	1	0	1	0	1	0	0	1	0	1	0	0	1	1

16bitマイコンの場合
　2バイト(16bit) ＝ ワード(Word)
　4バイト(32bit) ＝ ロングワード（ダブルワード）

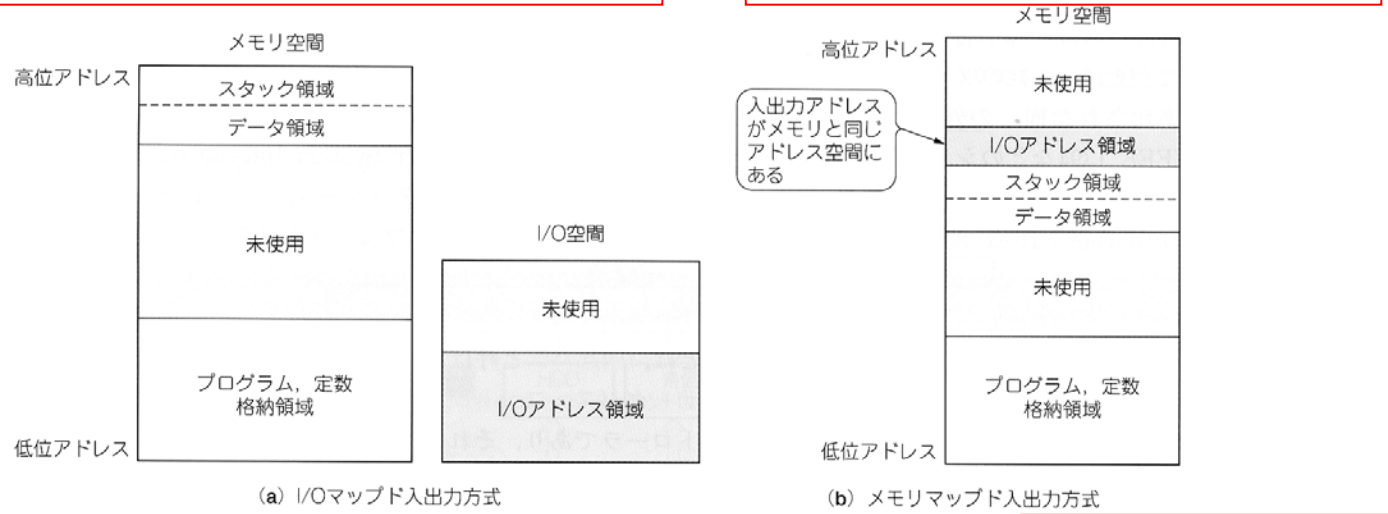


入出力方式

- I/O マップド入出力方式
  - 入出力命令があり、この命令により入出力対象となるポート番号を指定
- メモリマップド入出力方式
  - 入出力命令は無く、入出力デバイスのポートアドレスをメモリアドレス空間に配置し、通常のデータ転送命令（メモリへの書き込み・読み出し）によって入出力を実行

メモリアドレス空間を大きく使用可能

アドレス空間の自由度が高い



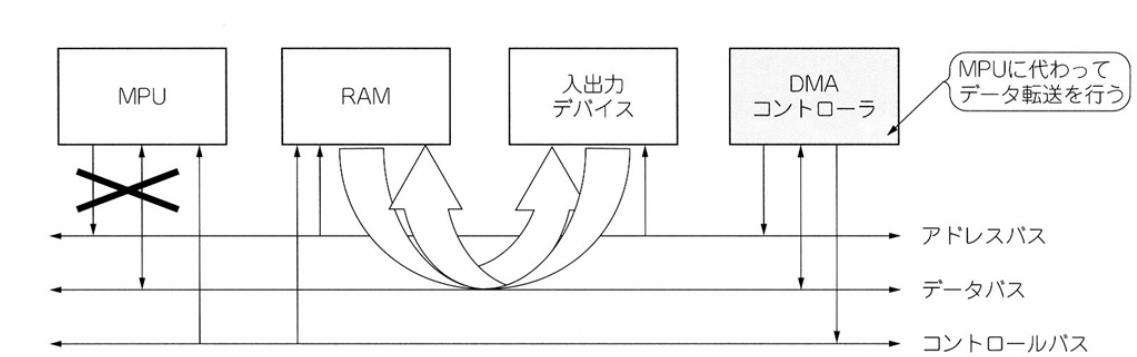
Intel系

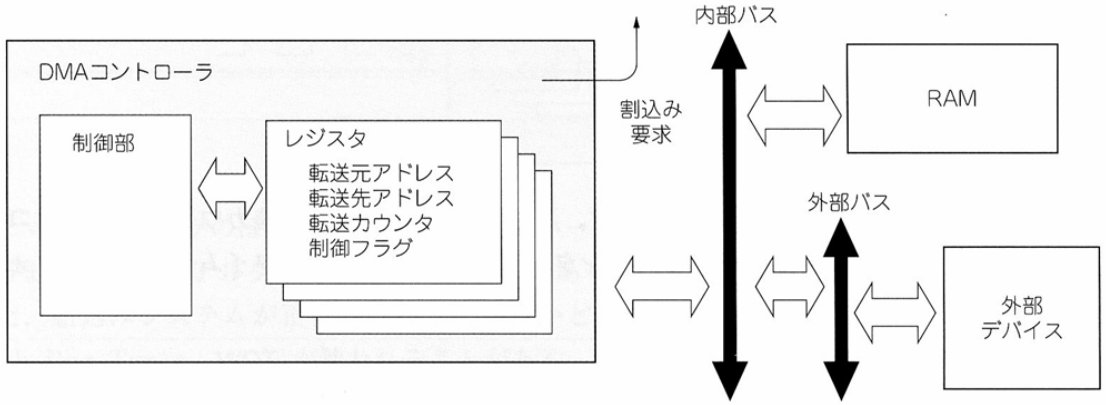
モトローラ系，RISC

I/Oマップドとメモリマップド入出力方式

DMA（Direct Memory Access）

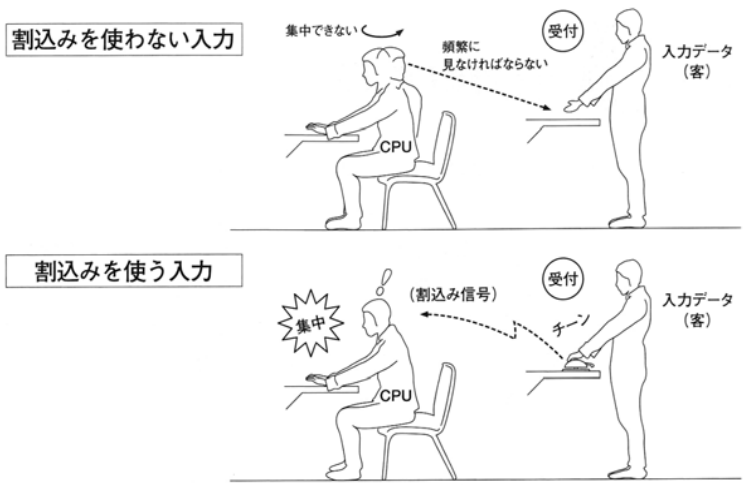
- 大容量データを扱う際にはCPUの負荷が多大
  ⇒　一時的にバスの制御機能をDMAコントローラに預ける
- CPUを介さず、入出力デバイスからメモリへ高速にデータ転送



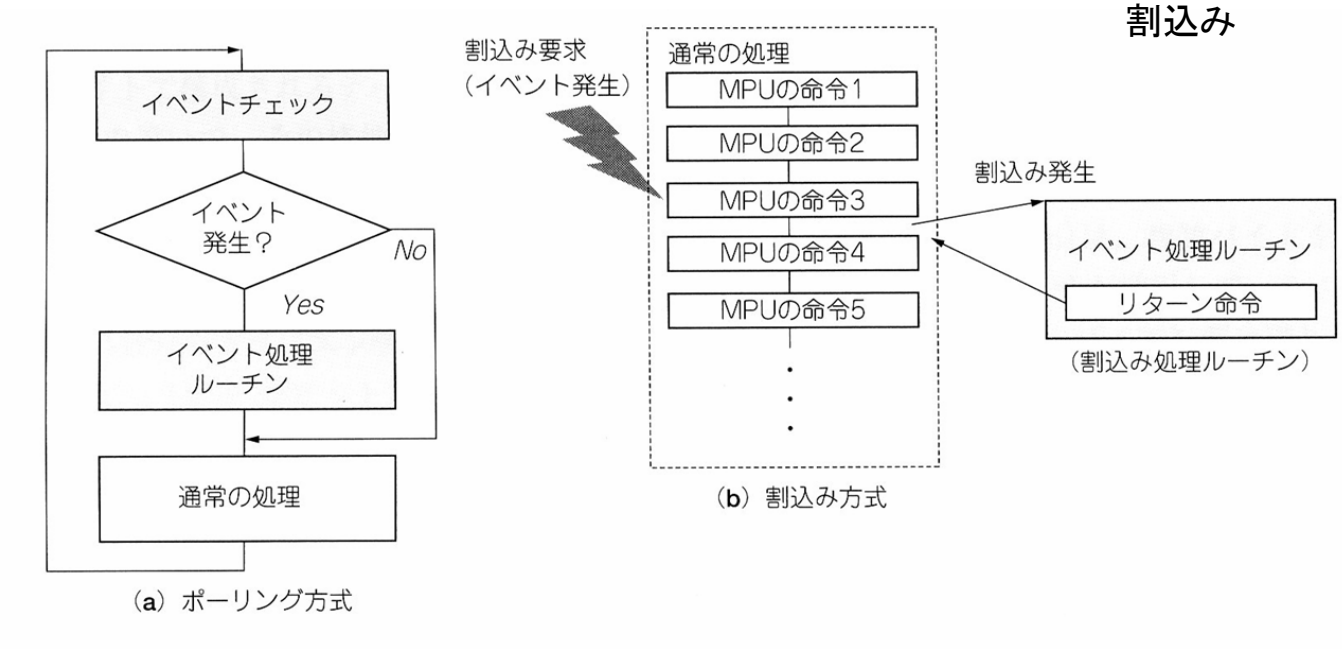


割込み

- 外部（内部）からの信号によって通常の処理を中断し、特定の処理を実行すること
  - PC（Program Counter）とPSW（Program Status Word）の内容をスタックに保存
  - 割込みの種類に応じたアドレスが指定された割込みベクタテーブルを解して割込み処理ルーチンを実行
  - 割込み処理ルーチンは通常処理で使っていたレジスタをスタック⇒終了したらレジスタを復元



割込み



割込みの種類

- 外部割込み
  - 入出力割込み
    - 入出力機器からの要求による割込み
      - ⇒ 入出力準備完了，入出力の正常な完了／異常な終了など
  - 異常割込み
    - 電源，メモリ，ソフトウェアの異常の検出による割込み
      - ⇒ ソフトウェア異常の検出・・・・WDT（Watch Dog Timer）を利用
- 内部割込み
  - ソフトウェア割込み
    - ソフトウェア割込み命令（トラップ命令）によって発生
  - 例外割込み
    - MPUが命令を実行する際に不正な命令コード，アドレス，演算などを検出したときに発生

WDTがオーバフローすると，WDT割り込みを発生したり，リセットをかけたりします．

割り込み処理

マイコンの主な割り込み要因

- タイマ
- 通信(UART,I2C,SPIなど)
- 汎用デジタルIOの変化
- AD変換終了
- リセット、パワーオン

イベントドリブン処理が可能

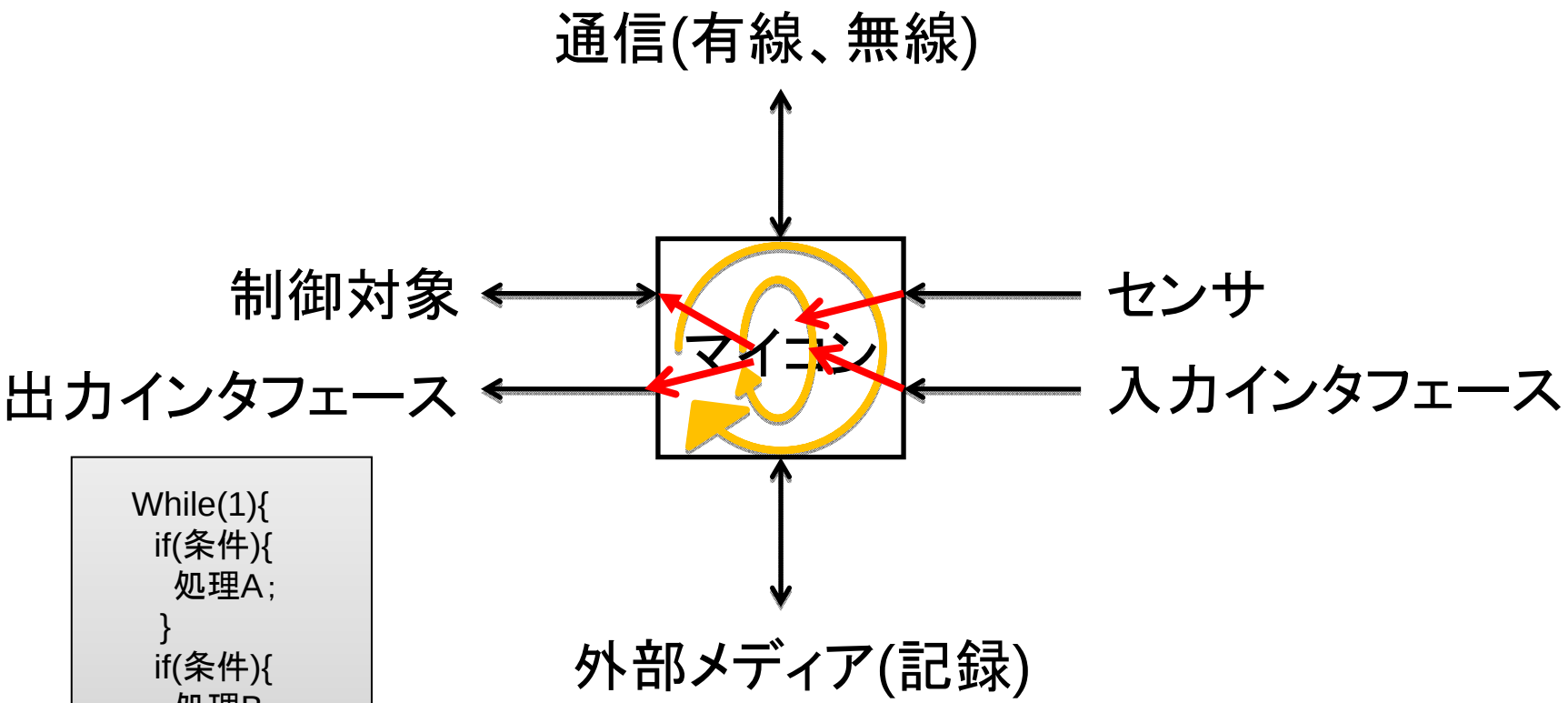
```
割込み要因1関数(){
    処理A;
}

割込み要因2関数(){
    処理B;
}

割込み要因3関数(){
    処理C;
}

int main(){
  スタートアップルーチン(){
      変数の初期化;
      周辺機能の動作設定;
      周辺機能のスタート;
      割り込み許可;
  }
  While(1){
        :
  }
}
```

割り込み処理




```
While(1){
  if(条件){
    処理A;
  }
  if(条件){
    処理B;
  }
  if(条件){
    処理C;
  }
      :
}
```

割り込みを使わない場合

ポーリング

タイマ／カウンタコントローラ

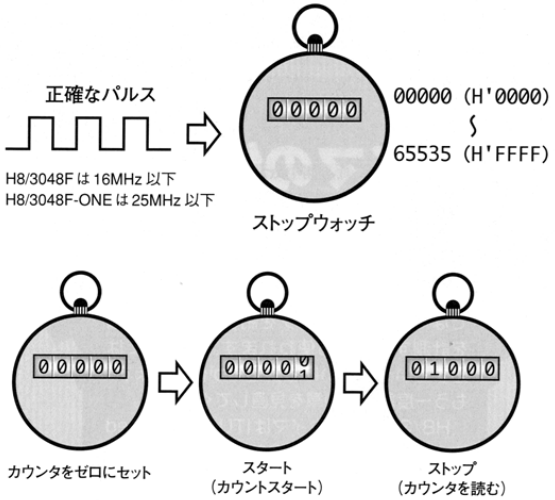
- タイマ　＝　パルスカウンタ
- カウンタコントローラ

  ⇒外部からのパルス入力が入る毎にカウンタを1つずつ加算（パルスを計数）
- タイマによる時間管理
  - トグル動作による一定周期パルスの出力

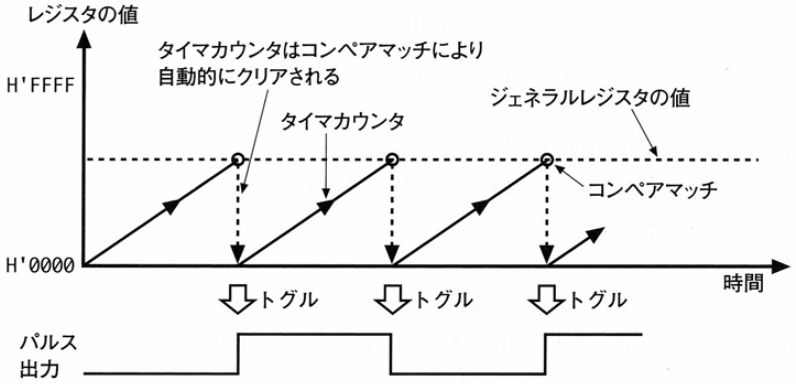
    ⇒　レジスタに設定したカウンタ値になるとタイマカウンタがクリア
  - PWM（Pulse Width Modulation）出力

    ⇒　周期一定でデューティ比を変化させたパルスを発生
  - インターバルタイマ

    ⇒　周期的な割込みの発生



タイマの動作



トグル動作

入出力回路(I／O)

- I／Oインターフェースの必要性

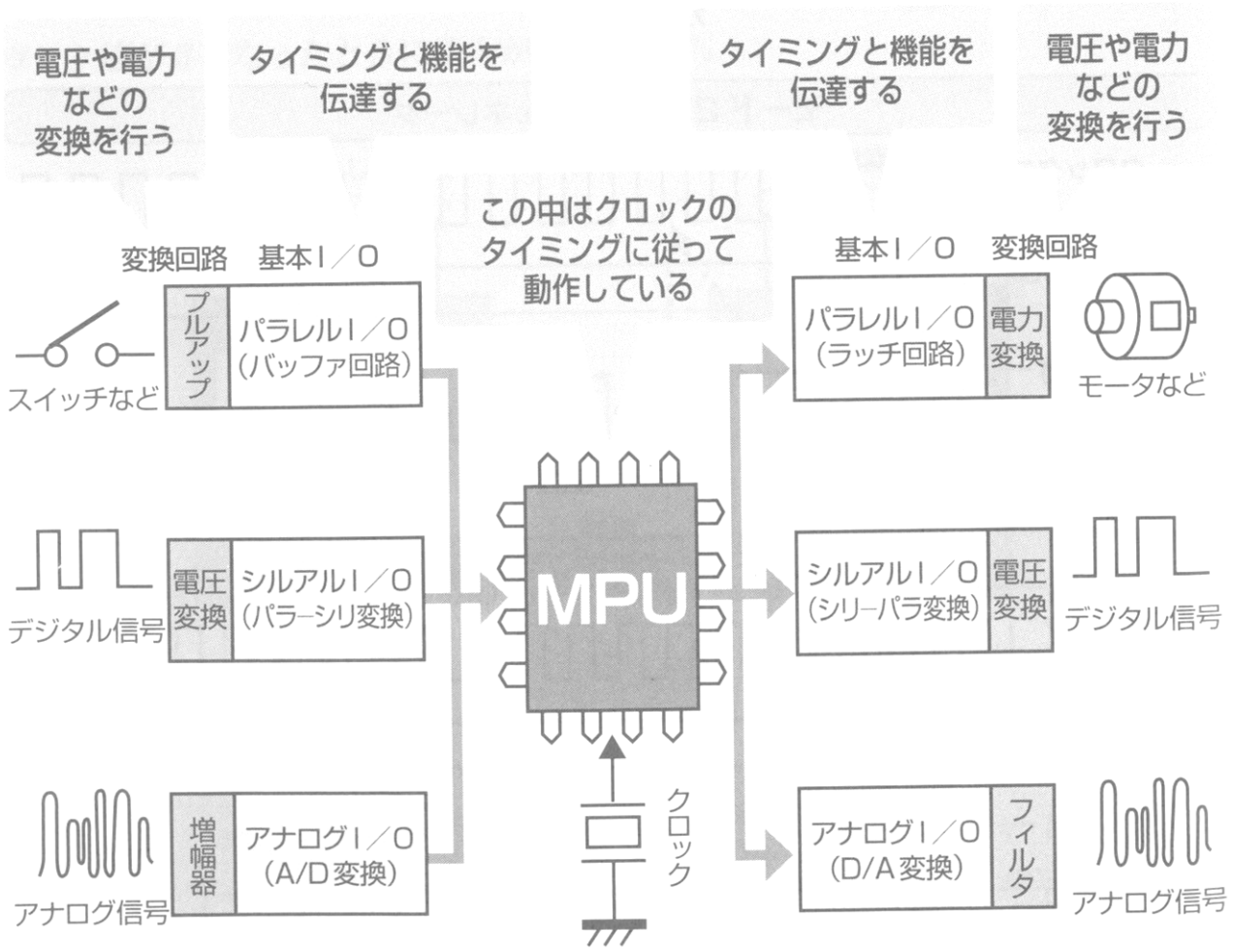
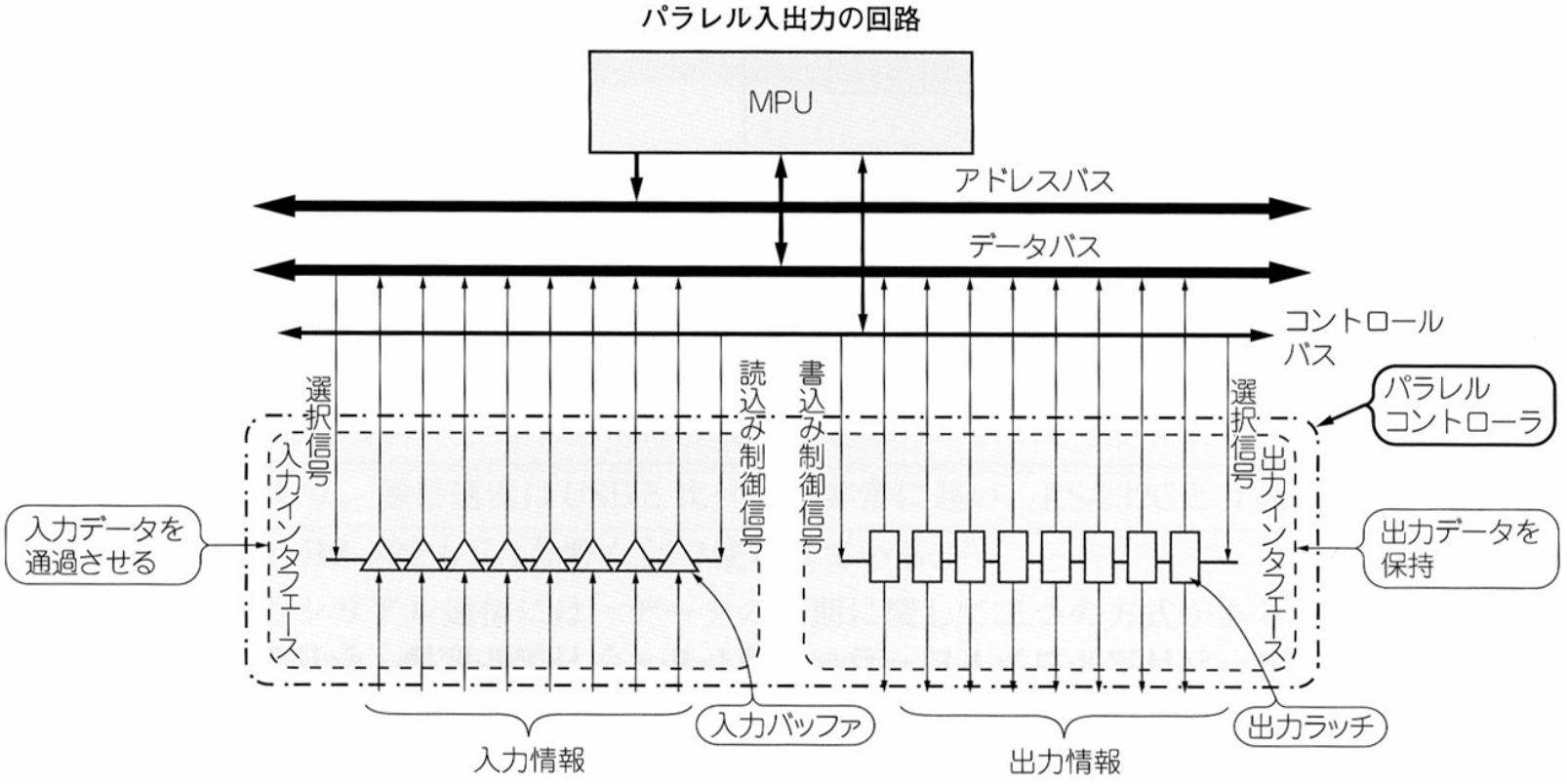


図 6-6-1　I/O インタフェースの構成

パラレルコントローラ

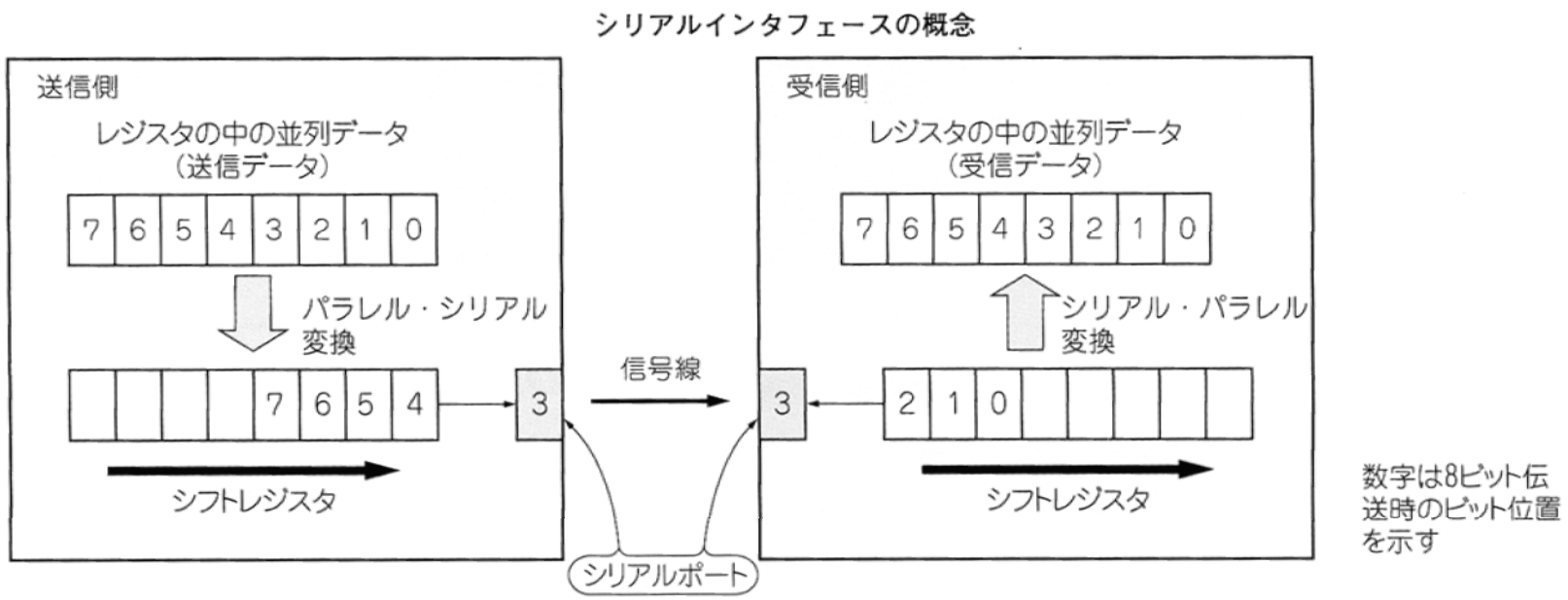
- パラレル入出力（Parallel I/O）
  - ⇒ 複数の信号線を使ってデータを並列（パラレル）に伝送
  - 入力バッファ ： MPUの入力命令実行のタイミングで信号を受け渡す
  - 出力ラッチ ： MPUの出力命令実行で出力された信号を保持し外部機器へ



パラレル入出力

シリアルコントローラ

- シリアル入出力（Serial I/O）
  ⇒ 1本の信号線に1ビットずつデータを直列に伝送



シリアル入出力の概念

⇒ USB（Universal Serial Bus）など広く使用

シリアル通信(USARTとRS232C)

・USART

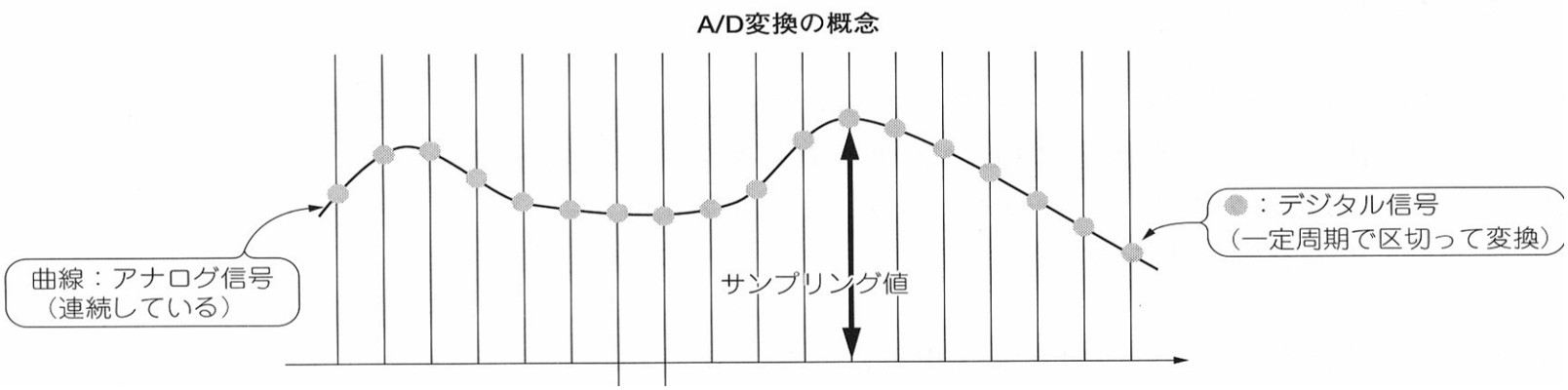
非同期式、同期式どちらも可能な汎用シリアルコントローラ。
ATmega644にはUSARTをハードウエアサポートする
モジュールが搭載されており簡単な手続きで外部機器
と通信ができる。信号レベルの違いを除きRS232Cとは
プロトコル互換。このため信号レベルの違いを変換する
IC(MAX232等)を利用することで、パソコンとの通信も
簡単に実現できる。

・RS232C(ANSI/EIA/TIA-232-E)

デジタル機器同士を接続する汎用シリアル通信規格。
パソコンではUSB登場以前の外部機器接続用ポート
として、この規格のサブセットの一つ9ピンインタフェース
が多く採用されていた。機器を1対1で接続することを
前提に非常にシンプルなプロトコルで実装されている
ため簡単に扱える。

アナログ・デジタル変換（A/D変換）

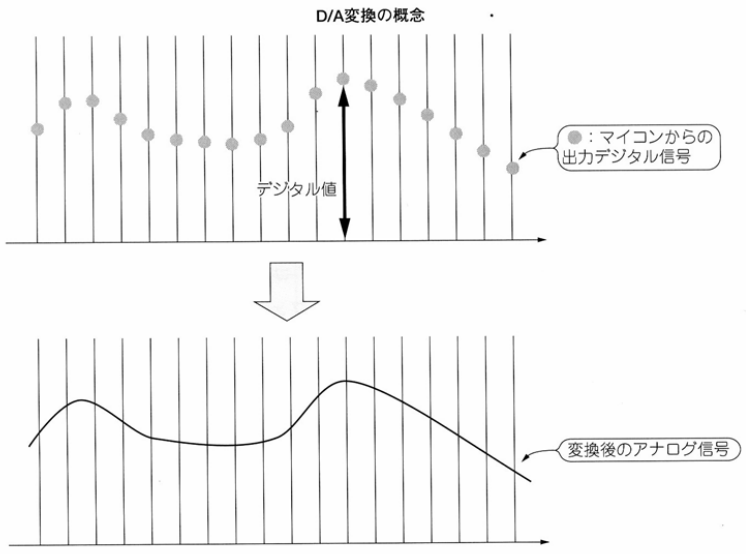
- 自然界の物理量（温度，音声など）はアナログ量（連続量）
  ⇔　マイコンで扱う信号はデジタル信号（離散量）
  - サンプリング周期毎にアナログ量をデジタル値に変換
    - サンプリング周期が短いほどアナログ値に近くなる
  - 分解能・・アナログ量を何ビットのデジタル値に変換するかで決定
    - 8ビットの場合・・　1／256
  - A/D変換方式
    - 逐次比較型，デルタシグマ型，パイプライン型など



A/D変換の概念

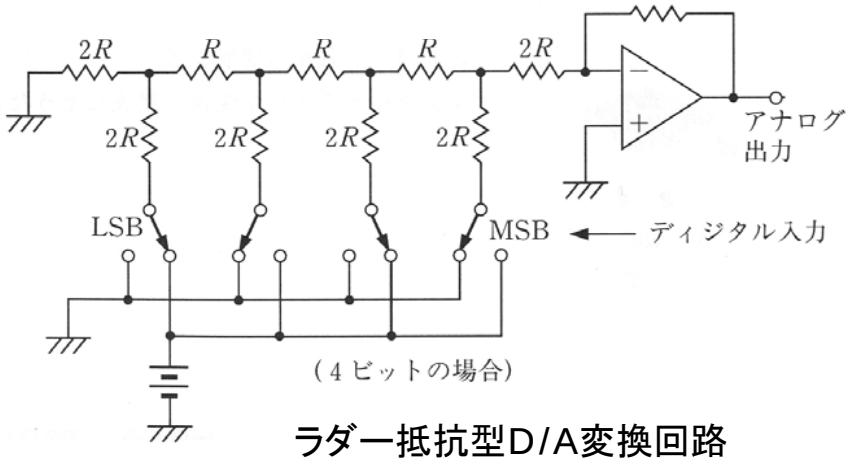
デジタル・アナログ変換（D/A変換）

- マイコンが処理した結果を外部に出力するために使用
  - アナログ・デジタル変換と同様にビット数が多いほど高分解能
  - デジタル値を設定してからアナログ値が安定するまでのセットリング時間（Settling time）を考慮する必要がある



D/A変換の概念

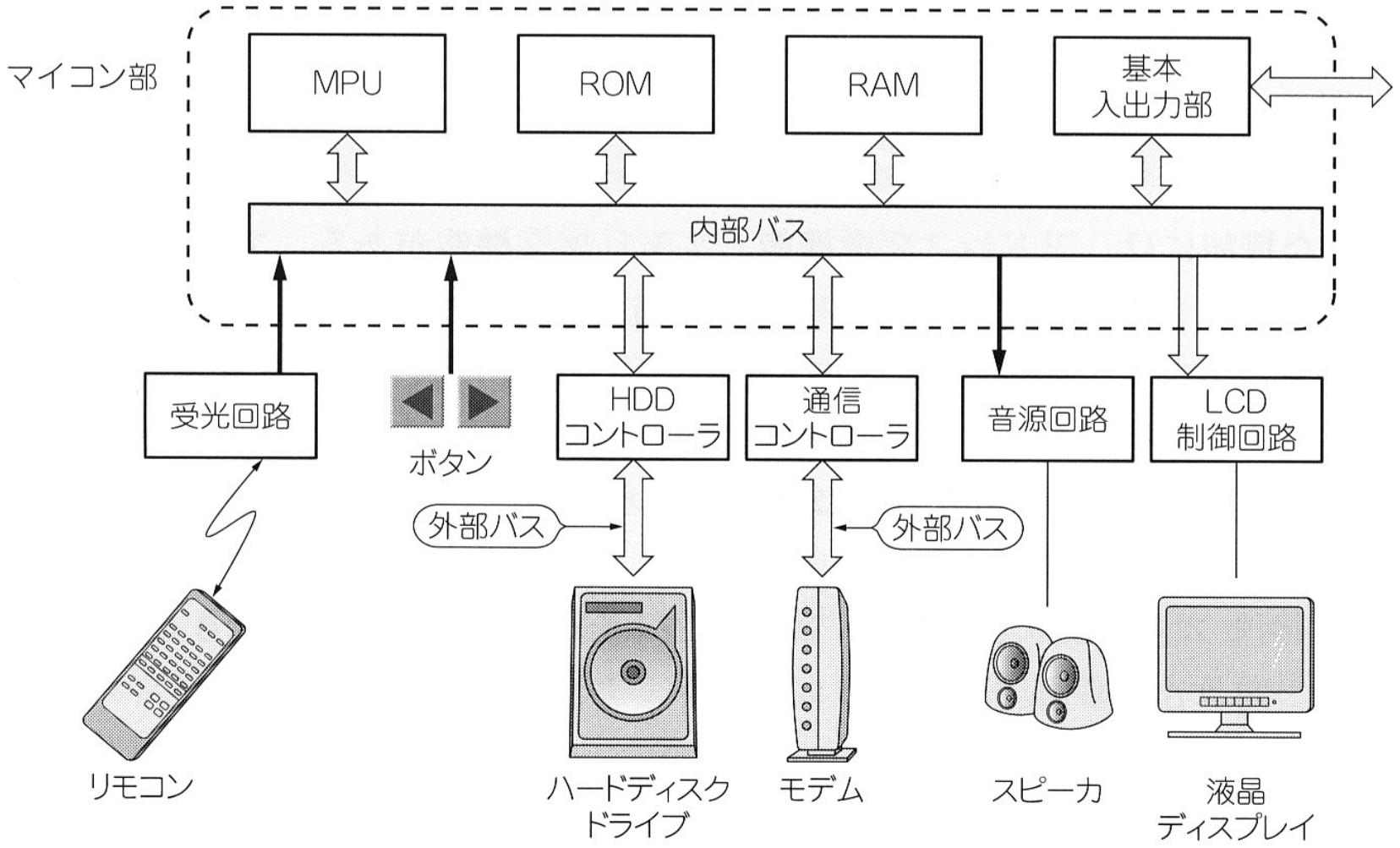
変換方式
- 重み抵抗型
- ラダー抵抗型
- デルタシグマ型
- PWM型



ラダー抵抗型D/A変換回路

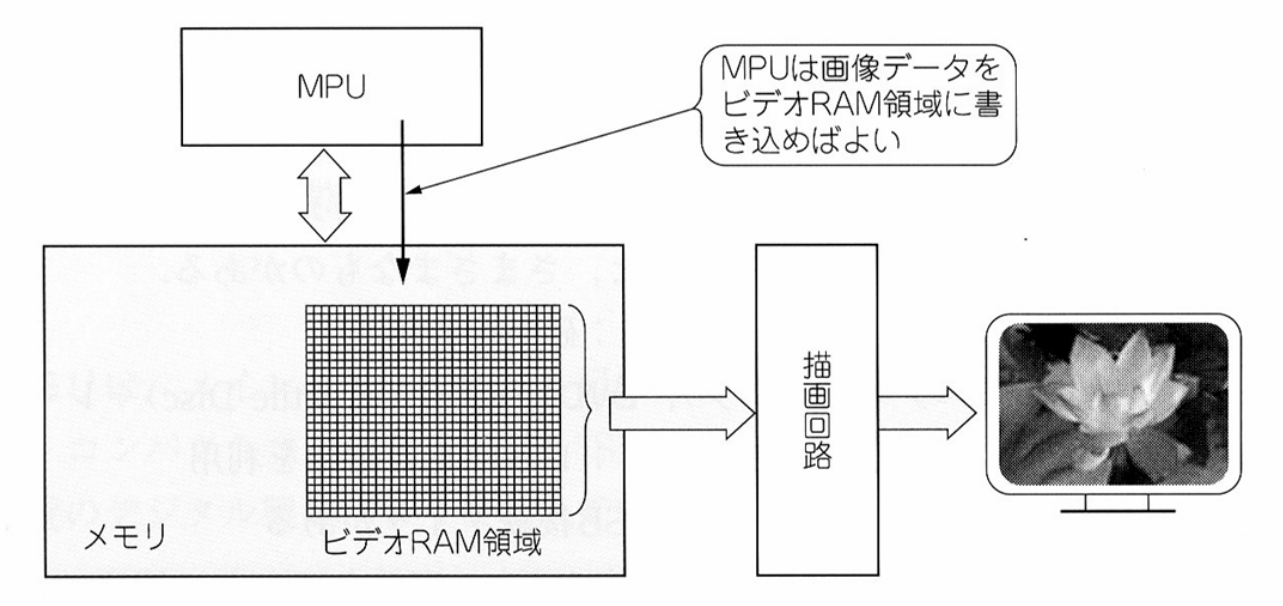
外部システム

- 入出力コントローラと周辺機器の間の情報伝達経路



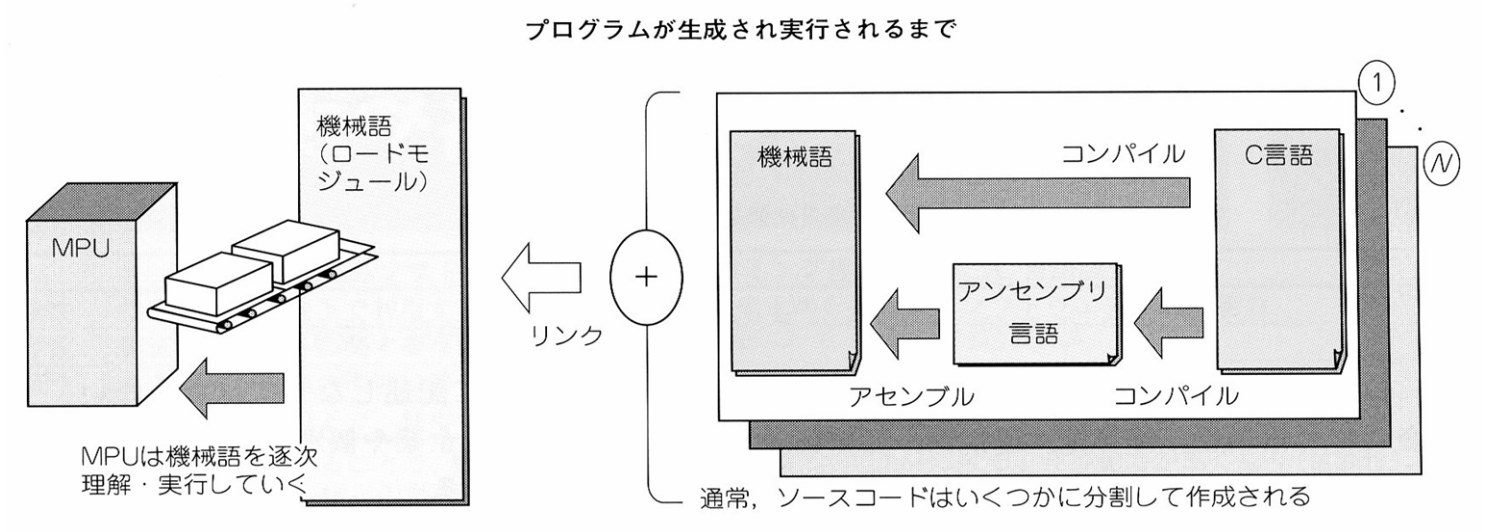
表示装置

- 組み込み機器からユーザへの情報伝達装置の一つ
  - LEDによるON/OFF情報
  - 7セグメントLEDやLCD（液晶ディスプレイ）などによる文字情報，画像情報
- LCDコントローラ
  - LCDへの表示を行なうためのLSI
  - MPUはVRAM（Video RAM）に表示データを書き込むだけ



プログラムの生成から実行まで

- プログラム
  ⇒ CPU(MPU)の処理手順を示すもの
- プログラム言語
  - 低級言語／低水準言語 ⇒ 機械語，アセンブリ言語
  - 高級言語／高水準言語（可読性が高い） ⇒ C言語など
    - コンパイル（Compile）： C言語 ⇒ （アセンブリ言語） ⇒ 機械語
    - コンパイラ（Compiler）： コンパイルを行なうソフトウェア
    - リンク（Link）： 分割してコンパイルされたオブジェクトコードを結合



引用元：「組み込みソフトウェア技術者試験クラス2対策ガイド」，(社)組込みシステム技術協会 編・著，CQ出版社，p.175

主なマイコン・メーカ

- ルネサステクノロジ　http://japan.renesas.com/
  - H8,R8Cファミリ　～　SHファミリ
- NECエレクトロニクス　http://www.necel.com/index_j.html
  - 78Kファミリ　～　V850ファミリ
- 富士通　http://jp.fujitsu.com/microelectronics/
  - F^{2}MCファミリ　～　FRファミリ
- テキサス・インスツルメンツ
  - MSP430ファミリ　http://www.tij.co.jp/jsc/docs/mcu/index.htm
- マイクロチップ・テクノロジ
  - PICマイコン
    http://www.microchip.com/stellent/idcplg?IdcService=SS_GET_PAGE&nodeId=64
- アトメル
  - AVRマイコン　http://www.atmel.com/JP/default.asp
- フリースケール・セミコンダクタ　http://www.freescale.co.jp/
  - HCS/8Q　～　ColdFireファミリ
- ARM
  - ARMプロセッサ・コア　http://www.jp.arm.com/

マイコンの選択ポイント

- 消費電力⇔処理能力
  リアルタイム処理の必要性の算定→クロック決定
  供給クロックと内部PLLで幅広い選択が可能
- 用途に応じた必要な機能の選択
  ワンチップ化(SoC)された機能の選択(機能一覧表やデータシート)
  低消費電力モード(**スリープモード**)
  ピン数、パッケージ、8ビットor16ビットor32ビット
- 電源電圧
  主流は5V系から3.3V系へ。
- コスト(価格、開発コスト)
  主流なパーツは安価だが、少し外れると高価で入手が難しくなる.
  ソフトウエア開発環境は無償、もしくは安価に手に入るようになったが、デバッグ作業に大きな進化は見られない。ハード化されたモジュールの利用や、同じマイコンを使ってコードや設計を流用することで開発や検証コストを抑える工夫が必要。

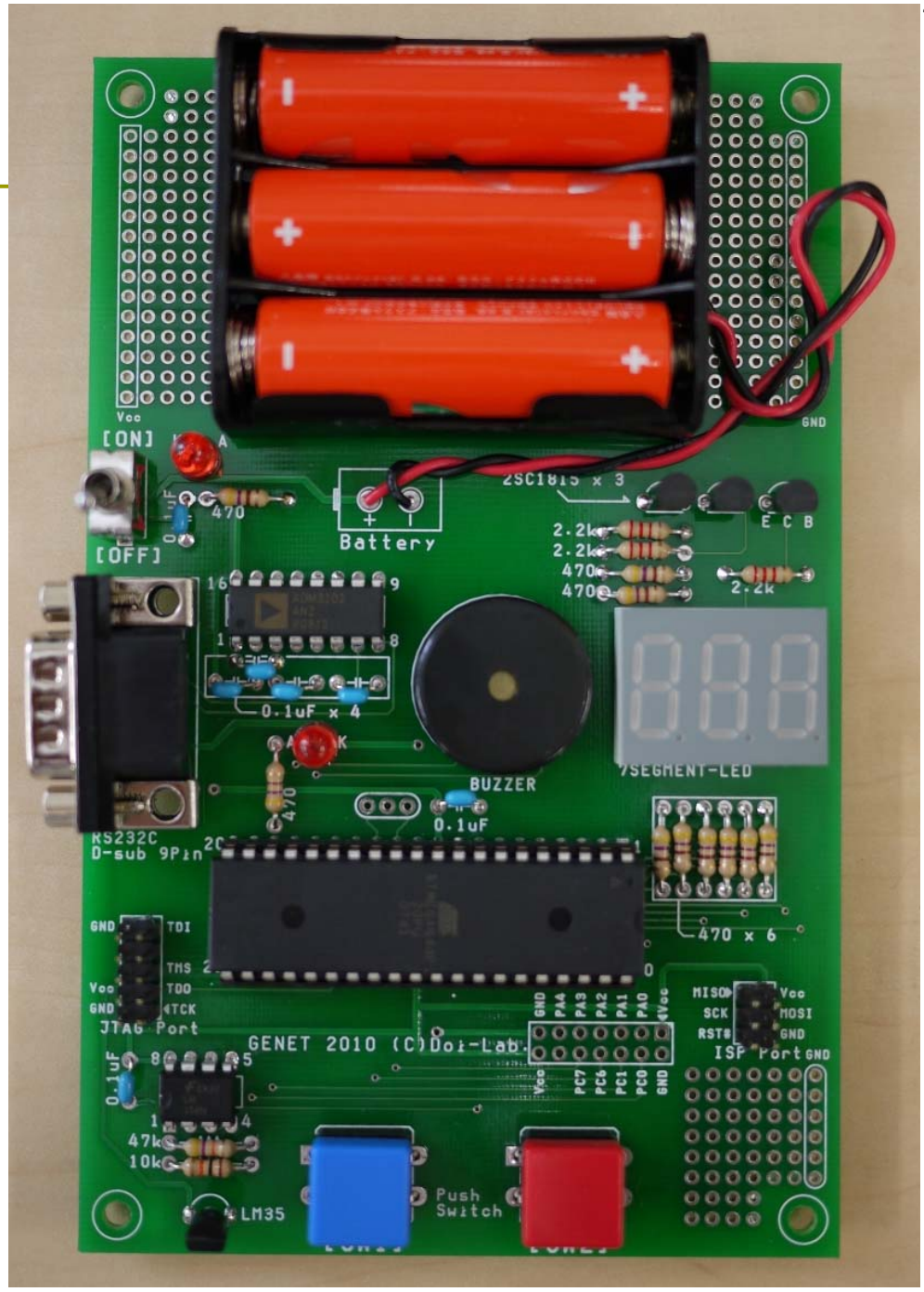
演習ボード製作

ワンチップマイコンボード

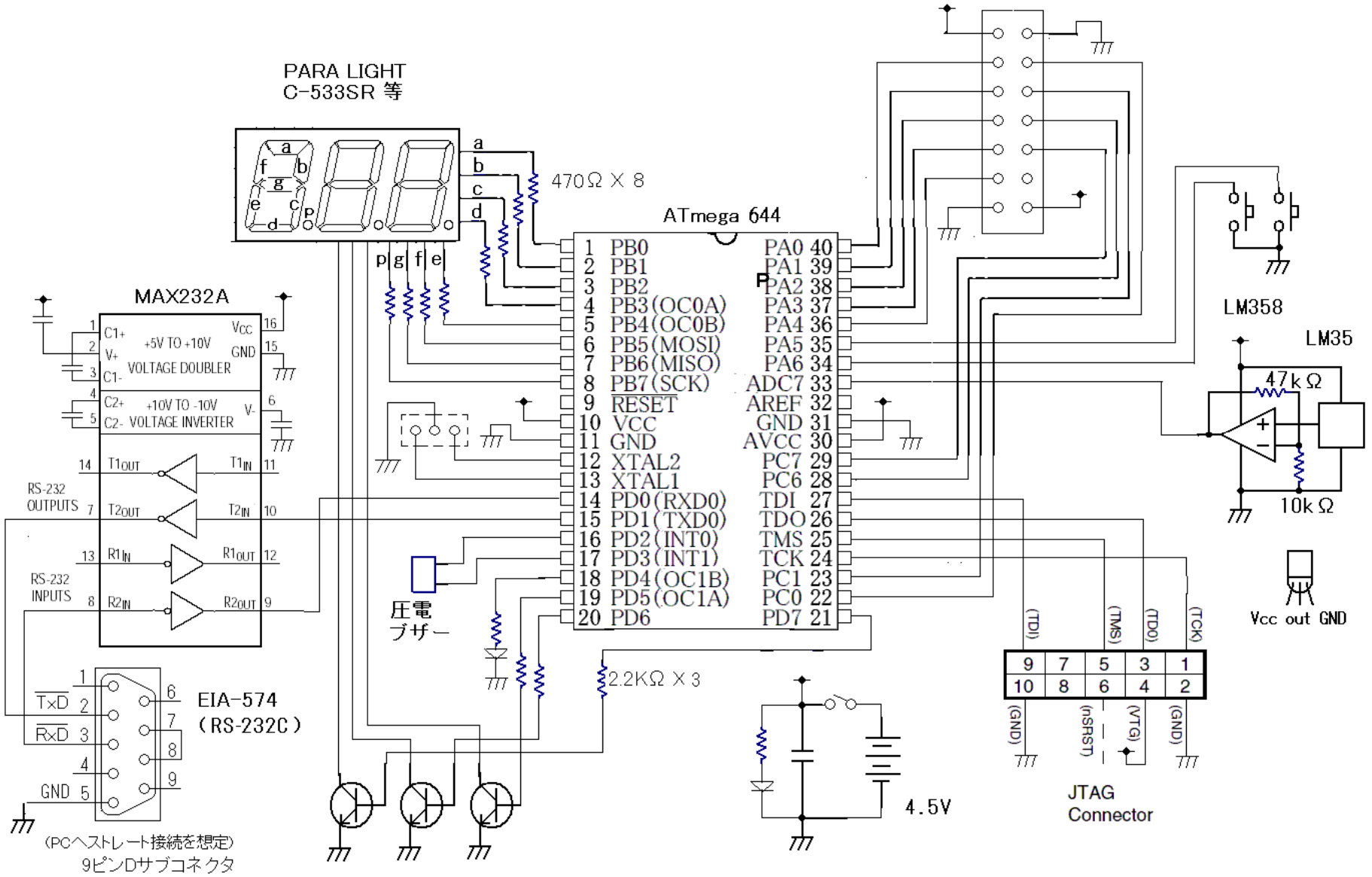
入力　:ボタンスイッチ
　　温度センサ

出力　:3桁7セグメントLED
　　ブザー

入出力:RS232C

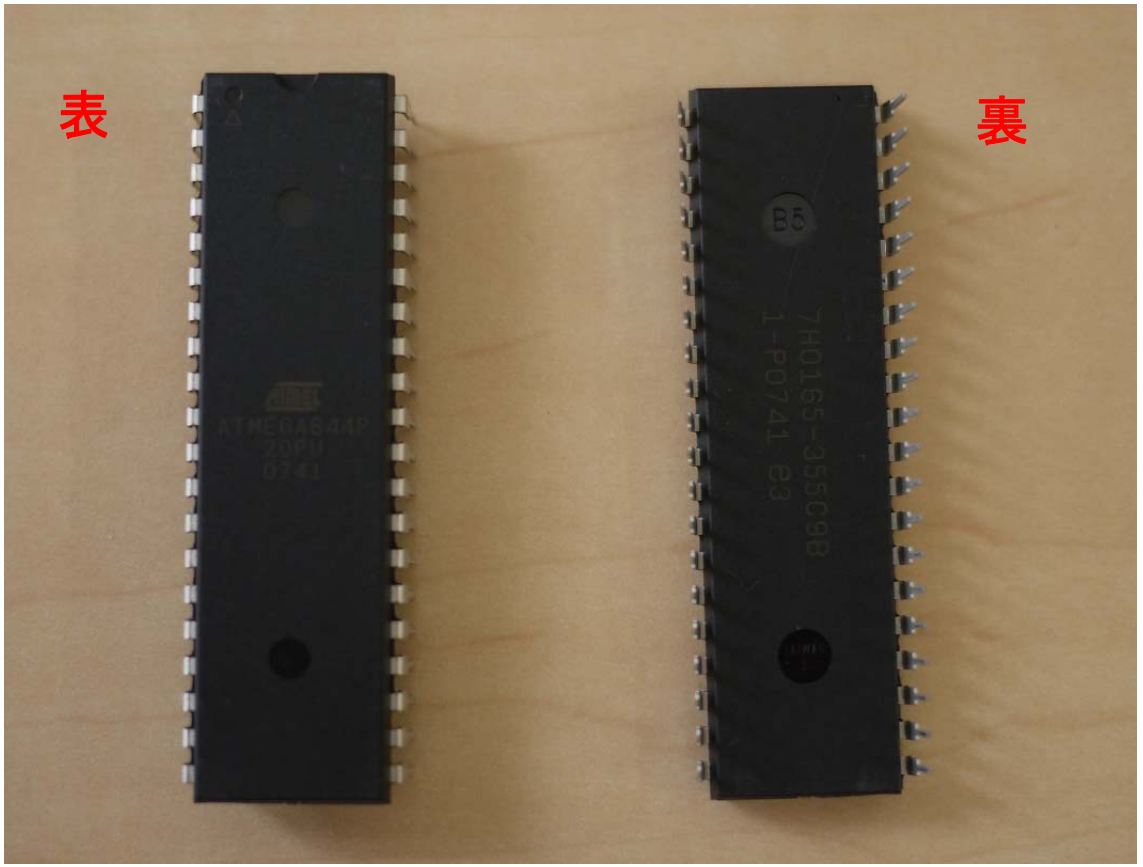


演習ボード回路図

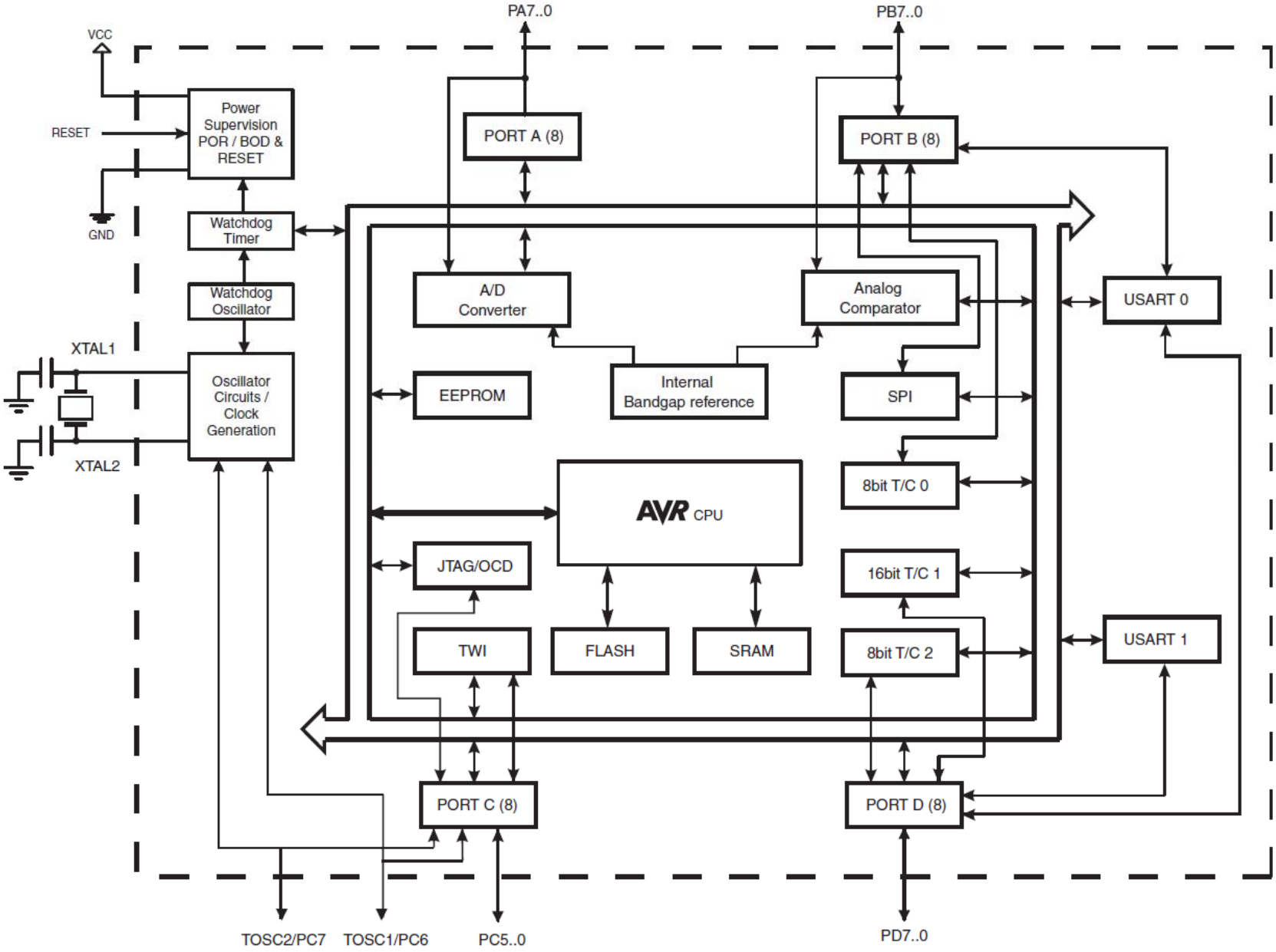


ATmega644P：ATMEL社製8ビットマイコン(RISC型CPUコア)　40ピンDIPパッケージ

ATmega644Pマイコンの外観(40PIN　DIPパッケージ)

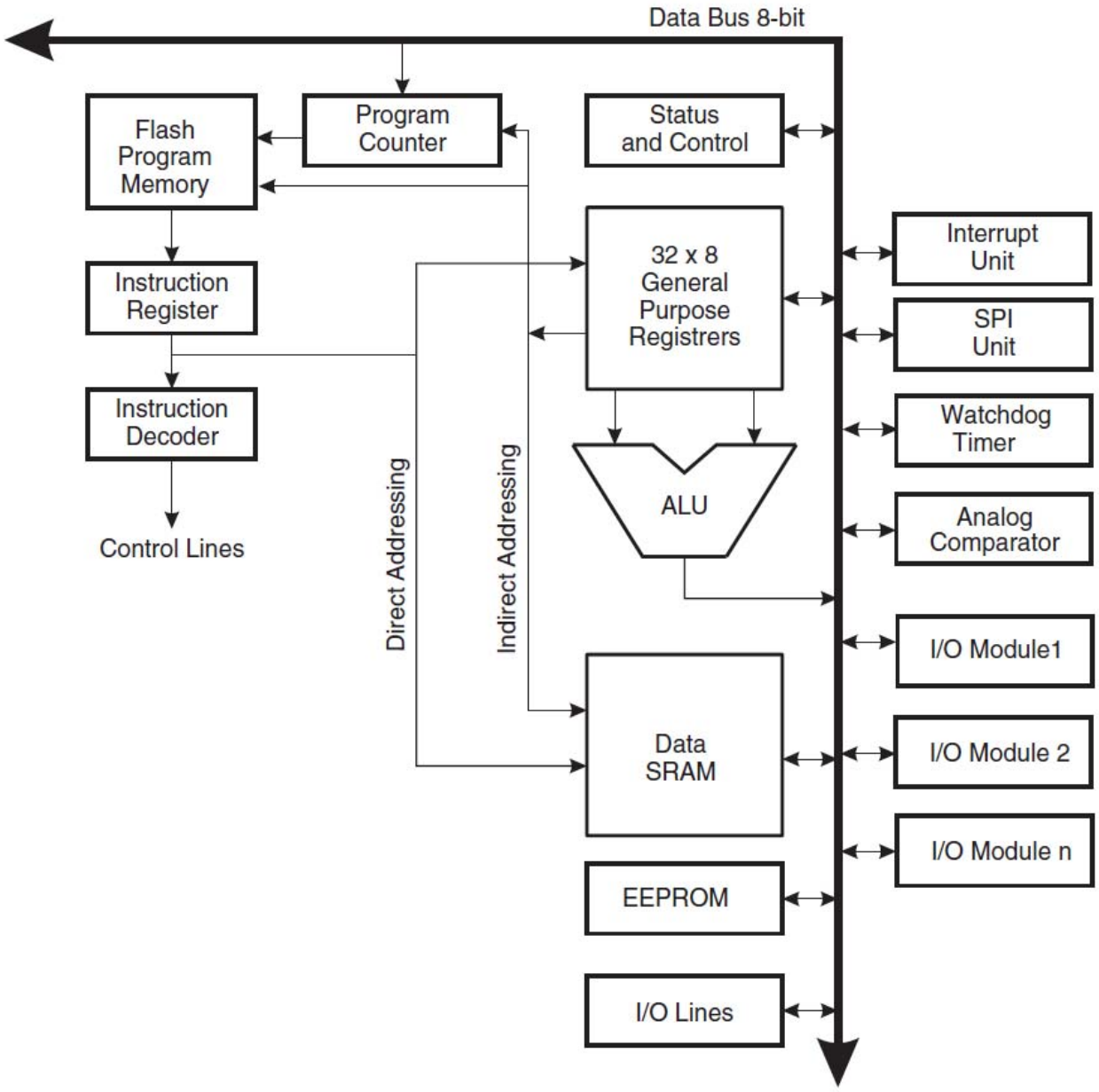


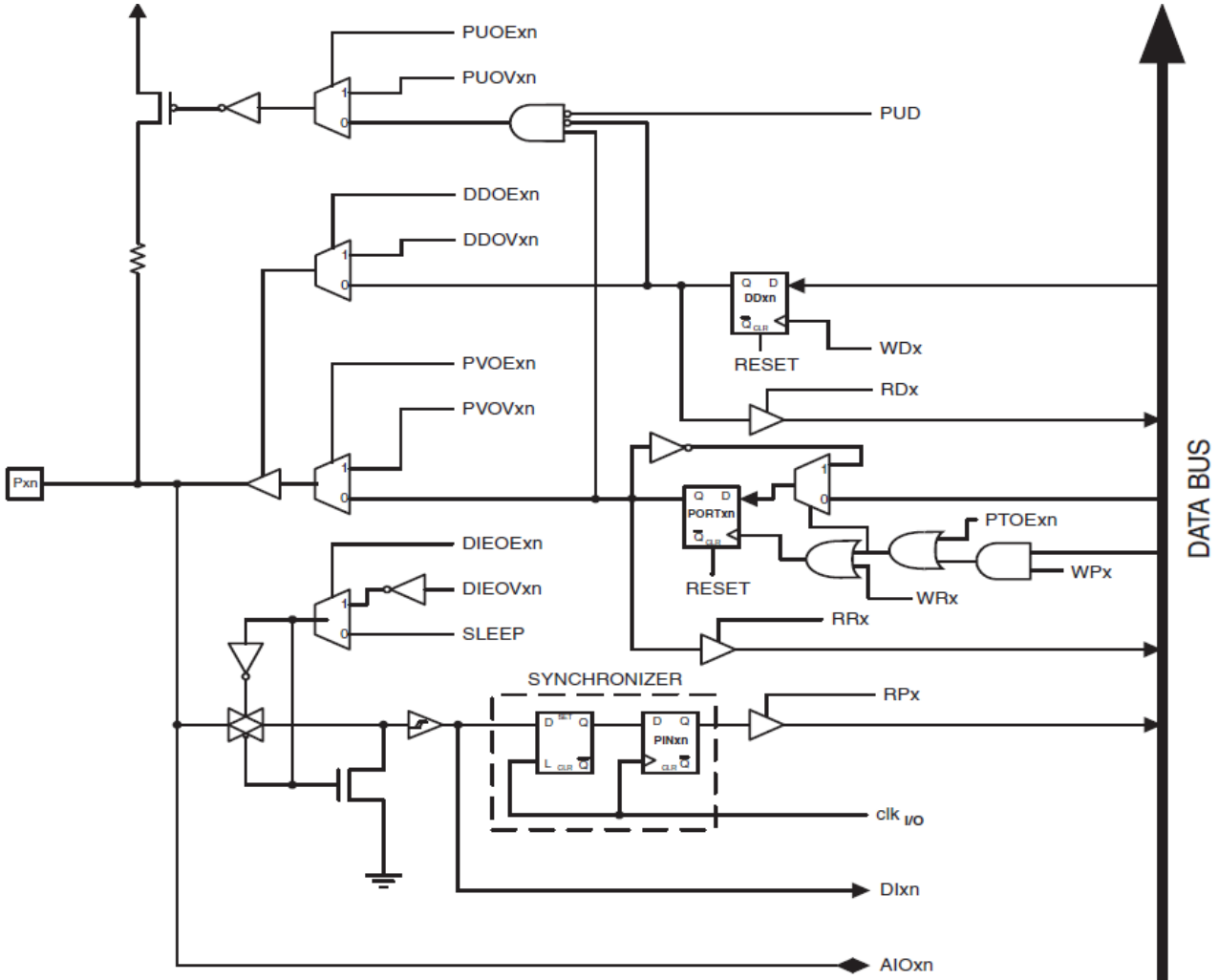
ATmega644Pマイコンはピンの電圧の変化で外部のデバイスと通信を行う。



AVR ATmega644Pマイコンの構成図

AVRの共通CPUコア





PUOExn:	Pxn PULL-UP OVERRIDE ENABLE
PUOVxn:	Pxn PULL-UP OVERRIDE VALUE
DDOExn:	Pxn DATA DIRECTION OVERRIDE ENABLE
DDOVxn:	Pxn DATA DIRECTION OVERRIDE VALUE
PVOExn:	Pxn PORT VALUE OVERRIDE ENABLE
PVOVxn:	Pxn PORT VALUE OVERRIDE VALUE
DIEOExn:	Pxn DIGITAL INPUT-ENABLE OVERRIDE ENABLE
DIEOVxn:	Pxn DIGITAL INPUT-ENABLE OVERRIDE VALUE
SLEEP:	SLEEP CONTROL
PTOExn:	Pxn, PORT TOGGLE OVERRIDE ENABLE

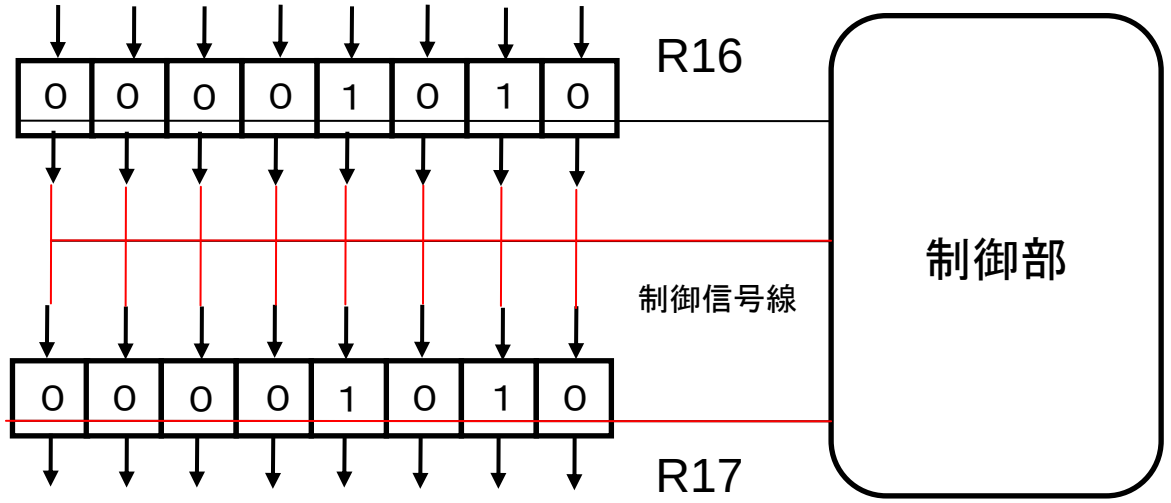
PUD:	PULLUP DISABLE
WDx:	WRITE DDRx
RDx:	READ DDRx
RRx:	READ PORTx REGISTER
WRx:	WRITE PORTx
RPx:	READ PORTx PIN
WPx:	WRITE PINx
$clk_{I/O}$:	I/O CLOCK
DIxn:	DIGITAL INPUT PIN n ON PORTx
AIOxn:	ANALOG INPUT/OUTPUT PIN n ON PORTx

AVRの兼用ピンの等価回路

CPUコアの動作例

動作例： MOV R17,R16

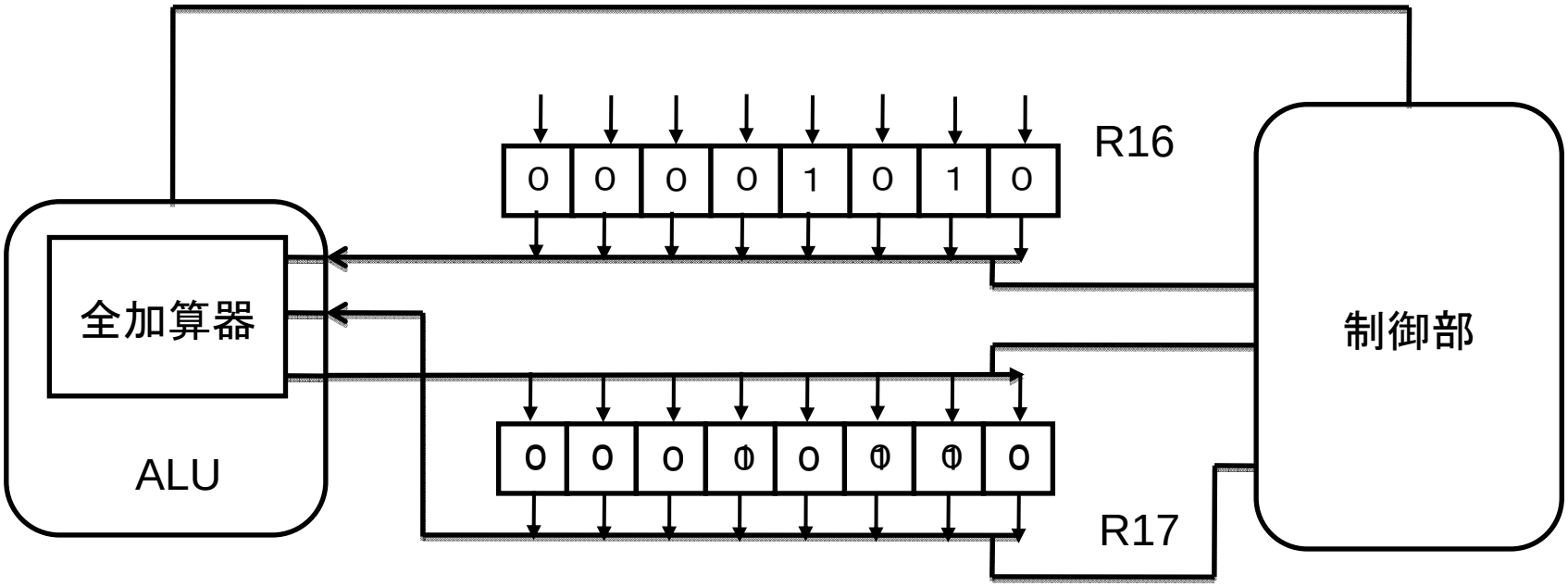
MOV命令を使って16番レジスタのデータを17番レジスタに移動させる。



CPUコアの動作例

動作例： ADD　R17,R16

ADD命令を使って16番レジスタのデータと17番
レジスタのデータを加算し17番レジスタへ格納する。

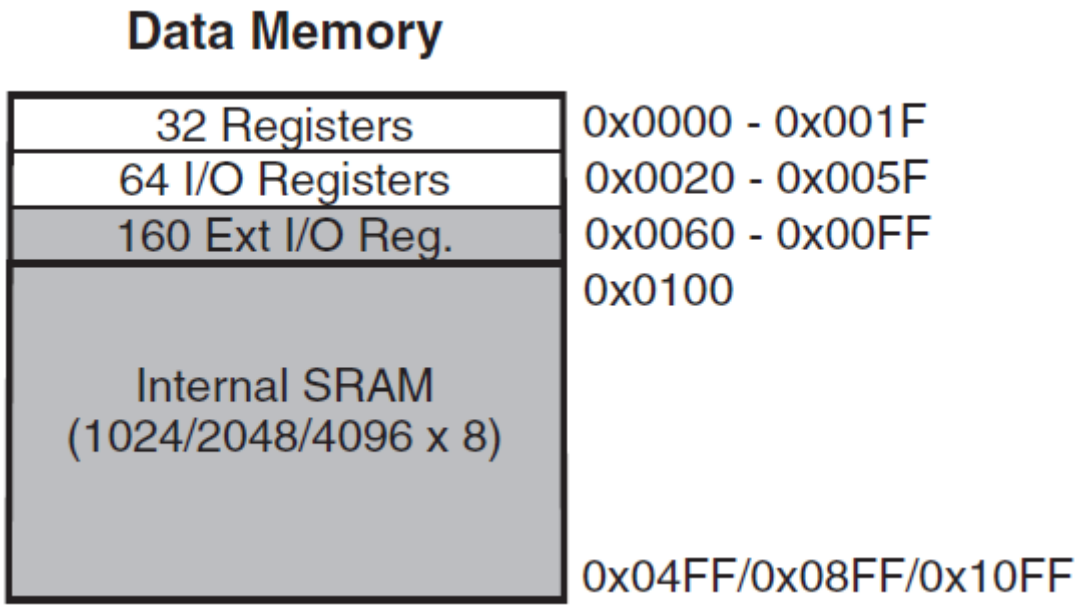


ワンチップマイコンの主な周辺機能

- タイマ、ウォッチドッグタイマ
  クロックをカウントアップすることで、時間指定でタイミング(イベント)を生成できる。
- AD変換器
  入力電圧をデジタル値に変換し、レジスタに格納する。
- シリアル通信(USART、I2C、SPI)
  汎用の通信プロトコルに準じたデジタルIO制御の基本的な操作を行う。
- JTAG又はISP
  外部との通信によりピンの状態をスキャンしたり、命令の実行を制御する機能。ICの製品検査やプログラムの簡易デバッグ、書き込みにも利用できる。ISPは書き込みのみ。
- EEPROM
  電気的に読書き可能な不揮発性メモリ。

メモリ空間

AVRのメモリ空間のアドレスは16ビットで表される。
プログラムメモリ空間とデータメモリ空間を別々にもつ
ハーバード・アーキテクチャ
I/O空間はデータメモリ空間に配置され、
メモリマップドI/O方式でアクセスできる。



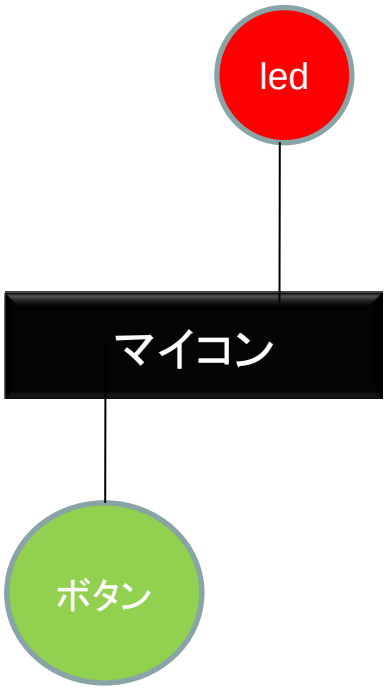
CPUの詳細（データシート）について

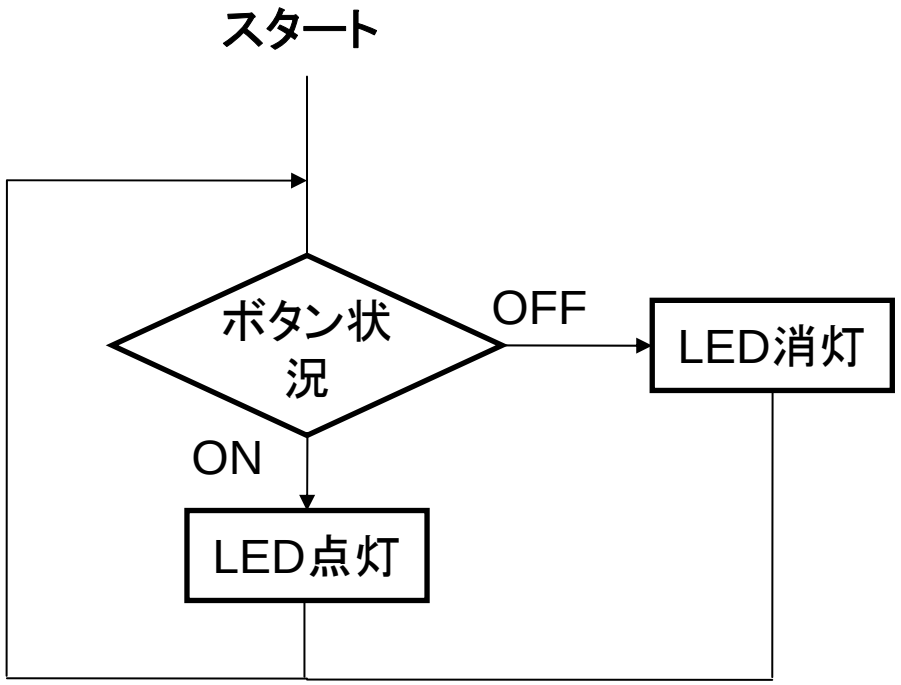
入手先：チップ供給元のホームページ
http://www.atmel.com/dyn/products/datasheets.asp?family_id=607#760
日本語データシートはユーザ有志によるサイトHERO'Sで公開(有償)されている。

今回利用するATMEGA644Pのデータシートは全376ページ。
しかし、同じシリーズのマイコンであればコアが共通のため
データシートの大部分は全く同じ。
更にメーカーが違っても、C言語で開発し、同クラスのマイコン
であれば、基本的な機能の差はあまり感じない。
違いは、コア周辺の専用ハード(ペリフェラル)の機能や性能、
ラインナップ一覧などで、用途に合わせてピックアップし、詳細
はデータシートで確認する。電子ファイルで入手し、チェックした
い項目を検索で見つける方法が有効。

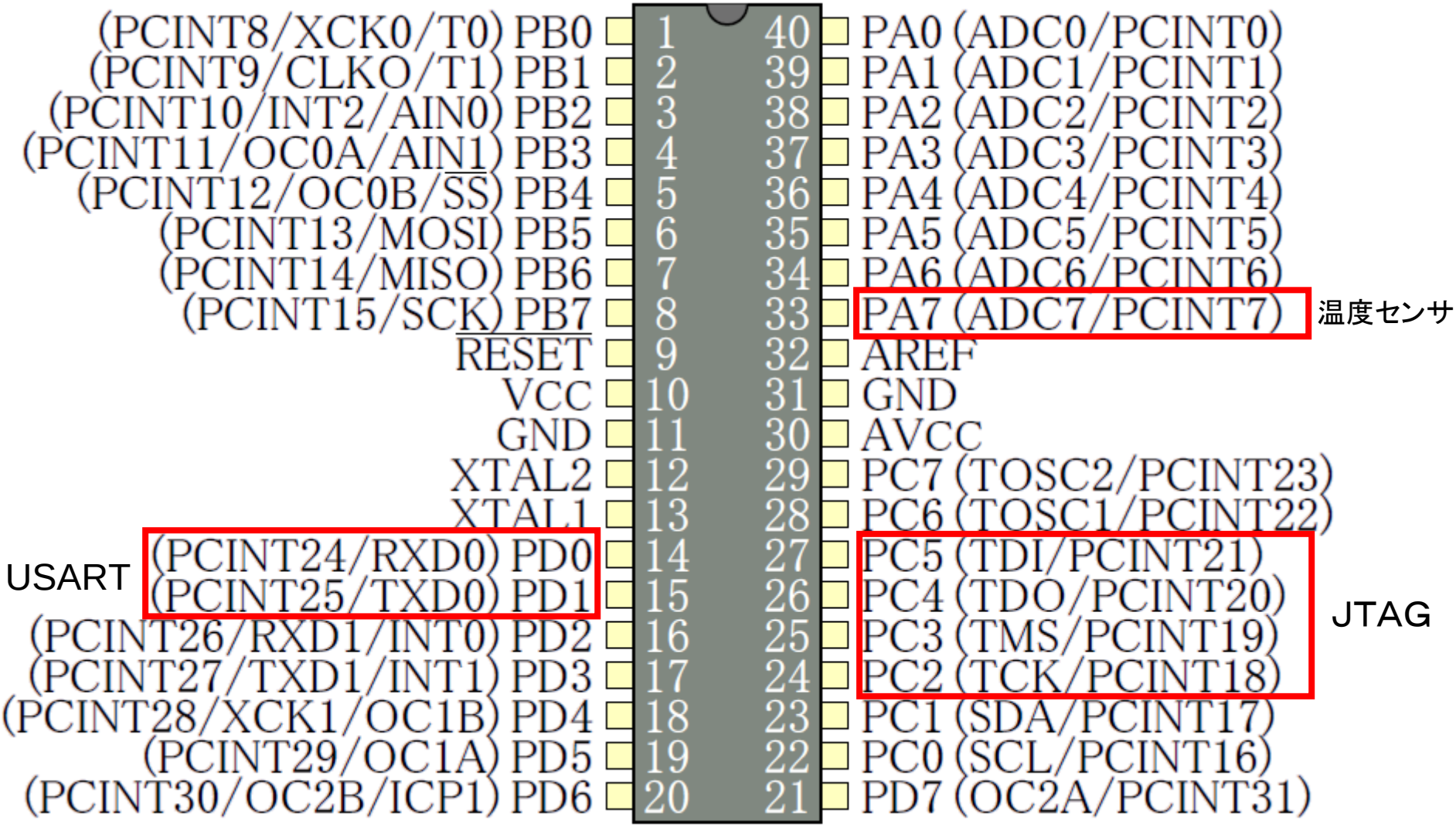
周辺装置との接続

周辺装置との接続はIOレジスタを介して行う。





AVR ATmega644pマイコンのピン配置



AVR　ATmega644pマイコンのピン配置

専用ピン

・VCC　電源供給ピン。電源の正極と接続。

・GND　グランドピン。電源の負極と接続。

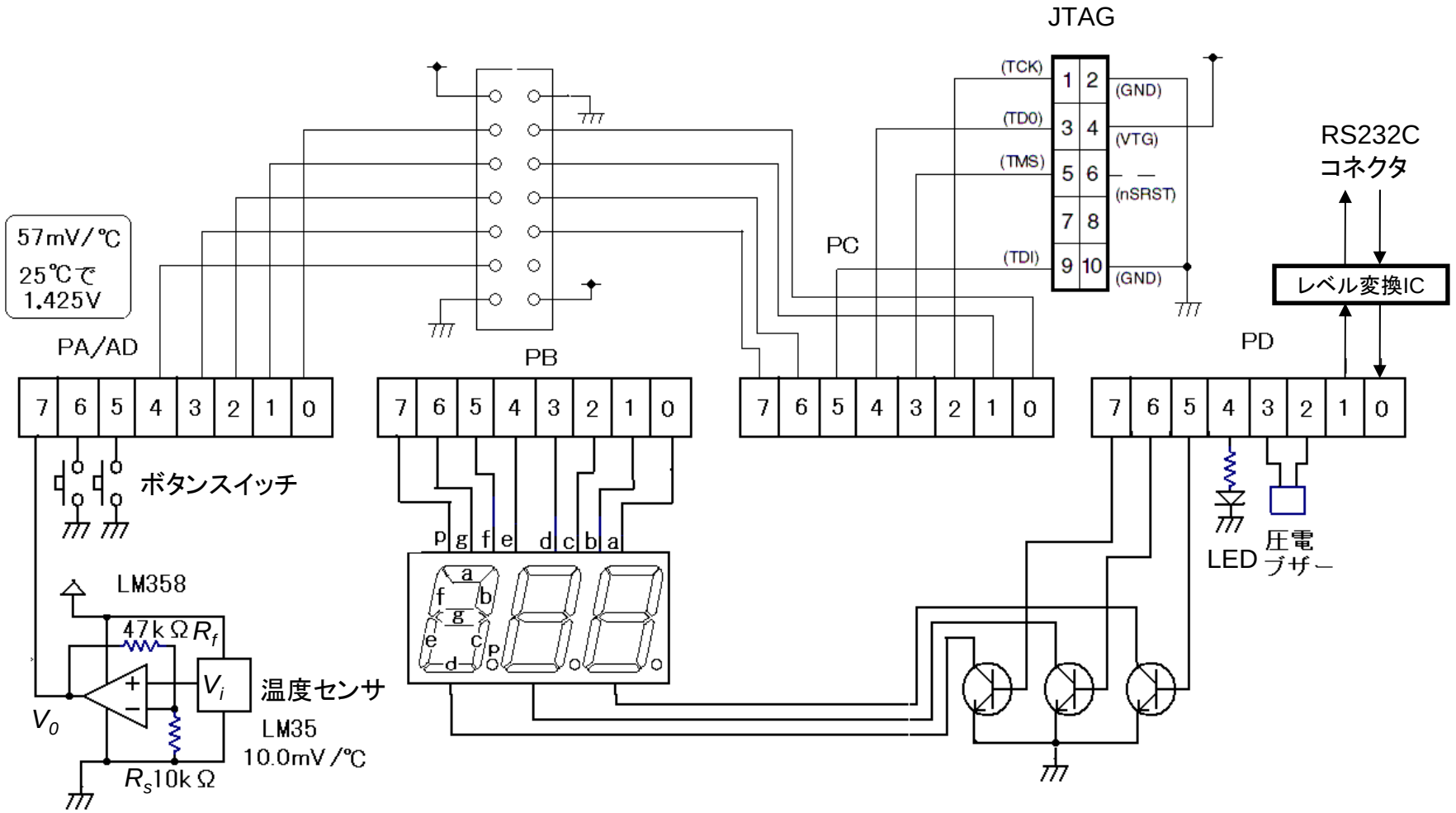
・RESET　2.5uS以上RESETピンがLOW(GND)になるとマイコンがリセット
(リスタート)される。

・XTAL1,XTAL2　外部発振子接続ピン。

・AVCC　内部AD変換器用供給電源ポート。電源の正極と接続。

・AREF　内部AD変換器用基準電源。AD変換器の計測上限電圧となる。
下限はGND。接続しない場合はVCCから供給された電圧が上限
電圧として採用される。

教材ボードのソフトウエアから見たブロック図

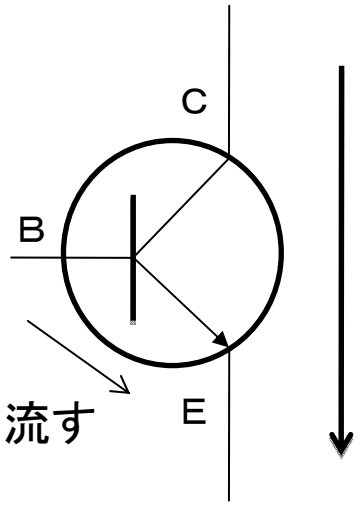


非反転増幅回路　　$V_0 = \dfrac{R_S + R_f}{R_s} \cdot V_i$

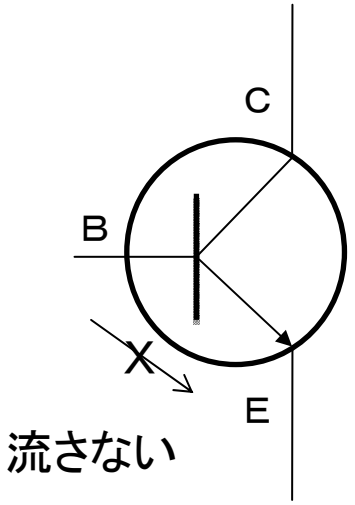
トランジスタ

電気で操作できるスイッチ

トランジスタ：ベース(B)エミッタ(E)間にわずかな電流を流すことでコレクタ(C)エミッタ間に電流が流れる。BE間の電流を流す、流さない(電圧をかける、かけない)を操作することでCE間の通電を制御するスイッチとしての効果が得られる。

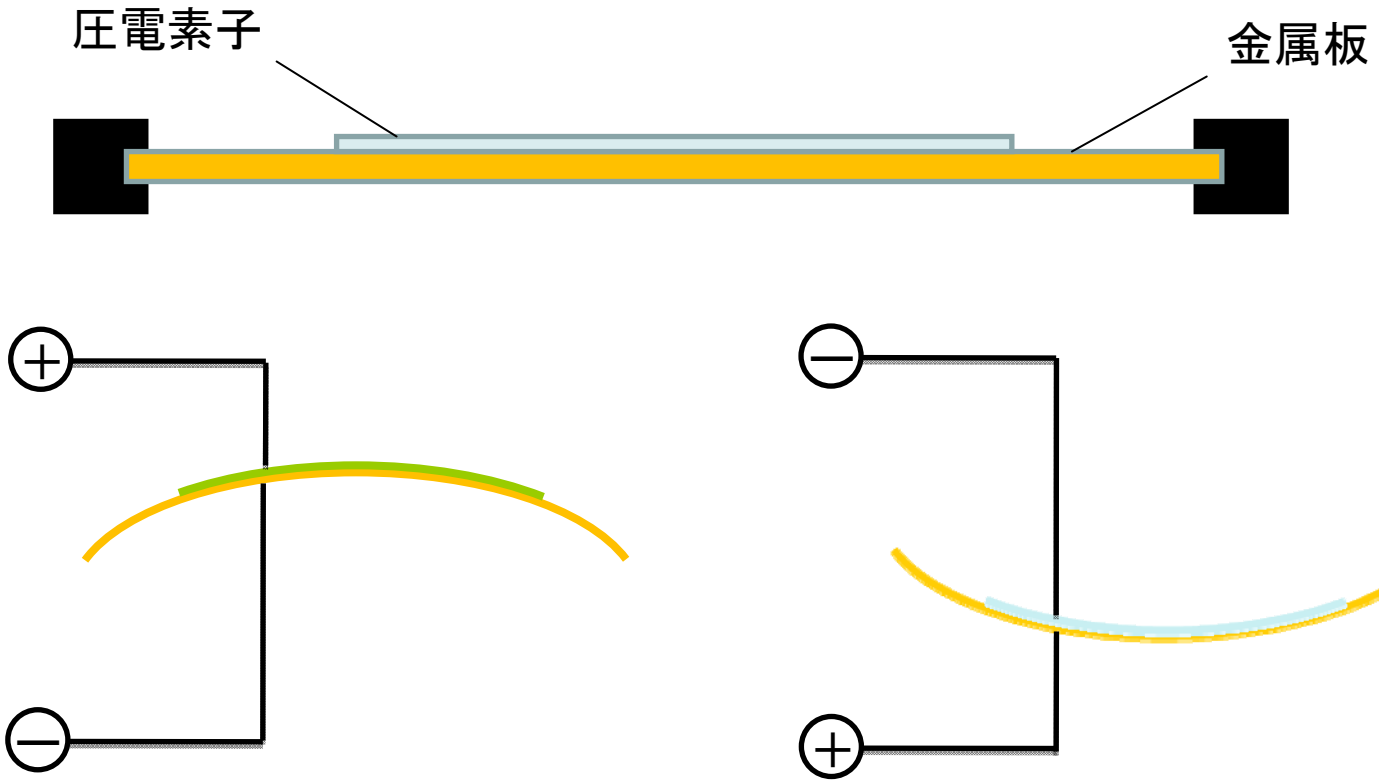


CE間に電流が流れる。



CE間に電流は流れない。

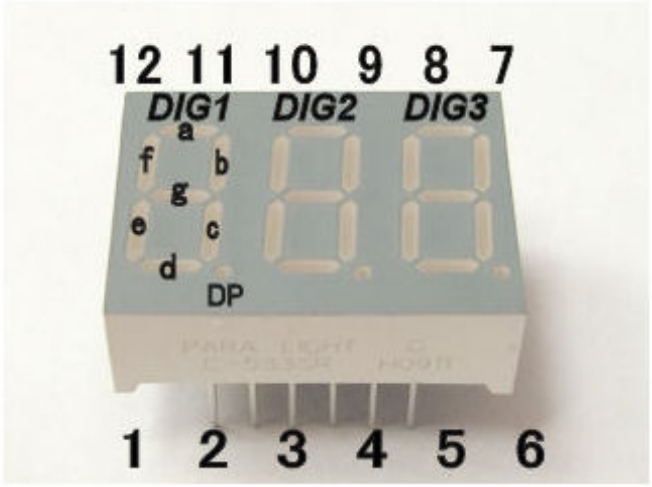
圧電ブザーの仕組み

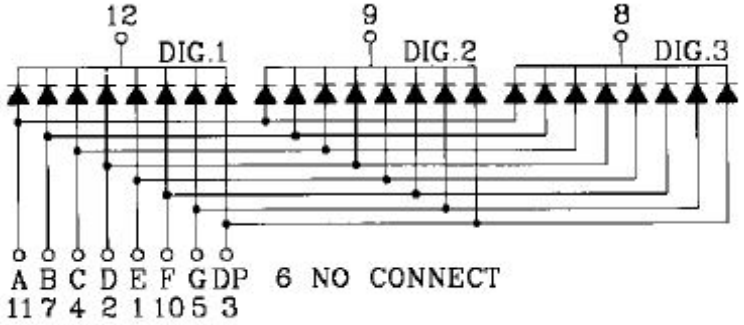


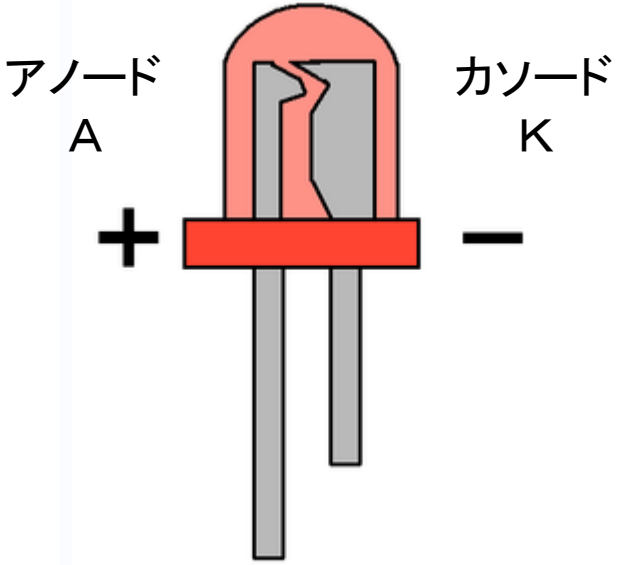
圧電素子が伸びて、金属板が山反りになる。

圧電素子が縮んで、金属板が谷反りになる。

部品リスト付録


12 11 10 9 8 7
DIG1 DIG2 DIG3
a
f b
g
e c
d
DP
1 2 3 4 5 6

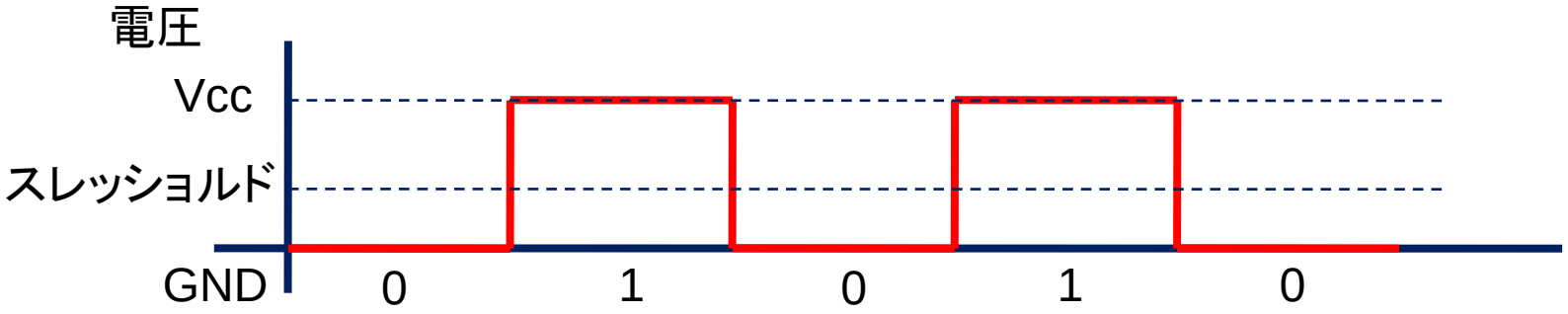

12
9
8
DIG.1
DIG.2
DIG.3
A B C D E F G DP 6 NO CONNECT
11 7 4 2 1 10 5 3


アノード
A
カソード
K
+
−

デジタル回路のアナログ要素1

- マイコンの入出力ピンの1(High)と0(Low)

  レジスタやメモリ上で0と1として扱う値はアナログ
  回路上ではHighはVcc,LowはGNDとして扱われる.



- クロック

  内部オシレータ回路で生成するか、または、外部
  発振回路(セラロックや水晶発振子)によって供給する.
  データの参照や書き込みなど, タイミングを合わせる
  ためのベースとなる. 一定であるのが望ましい.

デジタル回路のアナログ要素2

- バイパスコンデンサ

  ICへの電源供給を安定させるため、ICのすぐ近くに
  電源ラインVccとGNDを跨ぐように設けるコンデンサ。
  配線抵抗やICの消費電流の変化などにより起こる、
  供給電圧の変化を抑える目的と配線ラインに乗った
  高周波ノイズを除去する目的がある。0.01~0.1uFの
  セラミックコンデンサが良く用いられる。

- 保護抵抗

  過電流によるLEDの破損を防ぐため抵抗により電流量
  を制限します。オームの法則を使い次の式で求めます。
  (電源電圧 - LEDの順電圧) / LEDの順電流 = 保護抵抗値

ソフトウェア開発環境

- AVR JTAGICE Debugger mkII

バウンダリスキャンテストの標準方式

- AVR Studio 5
  - AVRマイコン用の統合開発環境(IDE)で書き込み及びデバックが可能
  - Windows XP/7で動作可能

３．開発環境の紹介

ソフトウエア

AVRには、アトメル社が無償提供する統合開発環境AVR Studio(SHやH8にはHEW、PICにはMPLABが用意されている)がある. C言語での開発も可能.

開発環境構築に必要なインストールソフト

AVR Studio (エディタ＋アセンブラ＋プログラマ+シュミレータ
C言語コンパイラ)

専用の統合開発環境を使うことで、ハードウエアの差異は選択や設定変更で吸収され、VisualStudioと同じような感覚で利用できるようになっている。

３．開発環境の紹介

プログラマ(書き込み機)

AVR Studioで作成した実行ファイル(＊.hex)をAVRのROM領域へ書き込む。

JTAG ICE mkⅡ

4. 実験ボード作成

部品リスト 1

部品名	メーカー	型番	極性	パッケージ	個数	記号	確認
プリント基板	―	―	―	両面2層	1	―	
電池ボックス	―	―	赤:＋黒:－	リード線	1	―	
ボックス固定用両面テープ	―	―	―	貼り付け済	2	―	
乾電池	―	―	あり	単3型	3		
マイコン	ATMEL	ATMEGA644P	ピン番号	DIP40Pin	1	―	
ICソケット(40P)	―	―	向きあり(窪み)	―	1	―	
RS232CインタフェースIC	Analog Devices	ADM3202	ピン番号	DIP	1	―	
オペアンプ	National Semiconductor	LM358N	ピン番号	DIP	1		
IC温度センサ	National Semiconductor	LM35DZ	Vcc-Out-GND	トランジスタ	1	―	
トランジスタ	東芝	2SC1815Y	E-C-B	トランジスタ	3		
赤色LED	―	―	あり(長い方が＋)	ピンタイプ	2		
Pinヘッダ10P(2x5)　JTAG	―	―	短い方を基板へ	ピンタイプ	1	―	
ICソケット(16P)	―	―	向きあり(窪み)		1	―	
トグルスイッチ(中点なし)	―	―	―	ピンタイプ	1		
Dサブコネクタ　9Pオス	―	―	ピン番号	ピンタイプ	1	―	

4. 実験ボード作成

部品リスト２

部品名	メーカー	型番	極性	パッケージ	個数	記号	確認
他励式圧電ブザー	KINGSTATE	KPE-813	―	ピンタイプ	1	―	
タクトスイッチ(カラーキャップ)	―	―	―	ピンタイプ	2		
7セグメントLED(3桁+ドット)	PARA LIGHT	C-533SR	ピン番号	ピンタイプ	1	―	
セラミックコンデンサ(0.1uF)	―	―	―	ピンタイプ	7		
抵抗(470Ω) 黄-紫-茶-金	―	―	―	ピンタイプ	10		
抵抗(2.2KΩ) 赤-赤-赤-金	―	―	―	ピンタイプ	3		
抵抗(10KΩ) 茶-黒-橙-金	―	―	―	ピンタイプ	1		
抵抗(47KΩ) 黄-紫-橙-金	―	―	―	ピンタイプ	1		
ゴム足	―	―	―	―	4	―	
はんだ	―	―	―	―	1	―	

カラーコード

黒：$0, 10^0=1$
茶：$1, 10^1=10$
赤：$2, 10^2=100$
橙：$3, 10^3=1,000$
黄：$4, 10^4=10,000$
緑：$5, 10^5=100,000$
青：$6, 10^6=1,000,000$
紫：$7, 10^7=10,000,000$
灰：$8, 10^8=100,000,000$
白：$9, 10^9=1,000,000,000$

金：$10^{-1}=0.1$ ±5%
銀：$10^{-2}=0.01$ ±10%
無： ±20%

４．実験ボード作成

はんだ付けについて

はんだ付けとは、はんだ材をはんだごてを使って熱で溶かし、
金属表面とはんだ材の結合を利用して部品基板に取り付けます。
電気的な接合だけではなく、取り付け強度とも関係しますので
丁寧に行ってください。目標は艶のある富士山型！！

はんだごての選び方

電子工作では、先が細めで手軽な20～30W程度のはんだごて
が扱いやすいです。鉛フリーはんだ対策には温度調節機能が
ついたものを利用し、300～350度程度の少し高めの温度
設定にすることで扱いやすくなります。また、ブーストボタンの
ついたもので一時的に高温を作り出すことでも対処できます。

4. 実験ボード作成

有鉛はんだと鉛フリーはんだ

はんだとして扱いやすい有鉛はんだが主流でしたが、環境への配慮から近年は鉛フリーはんだが利用されるようになっている。特に、RoHS指令などにより有鉛はんだを利用した製品の販売ができない地域もあります。

＊鉛フリーはんだは、有鉛はんだに比べ融点が20度ほど高いため、扱いずらい面があります。

RoHS指令とは

EU（欧州連合）が2006年7月1日に施行した有害物質規制。規制対象となっているのは，Pb（鉛），Cd（カドミウム），Hg（水銀），Cr6＋（6価クロム）PBB（ポリブロモビフェニル），PBDE（ポリブロモジフェニルエーテル）の 6物質。これらの物質が規制量を超えて含まれる電気電子機器は一部の除外対象を除き、EU地域での販売ができない。

4. 実験ボード作成

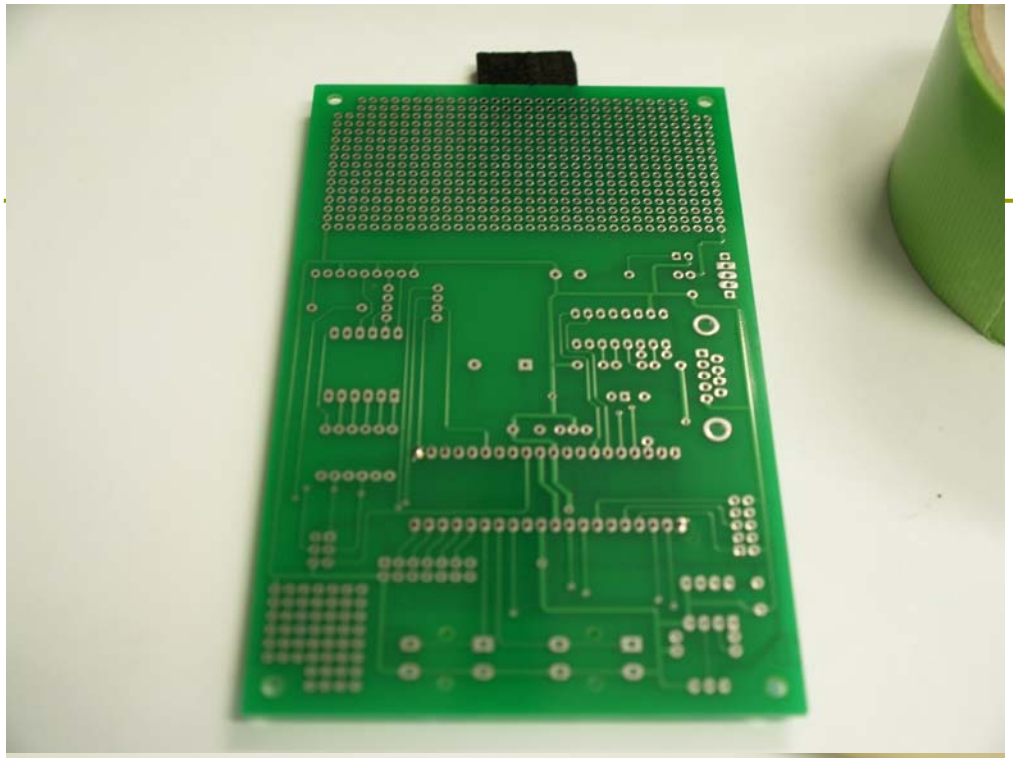
ボード作成における注意点

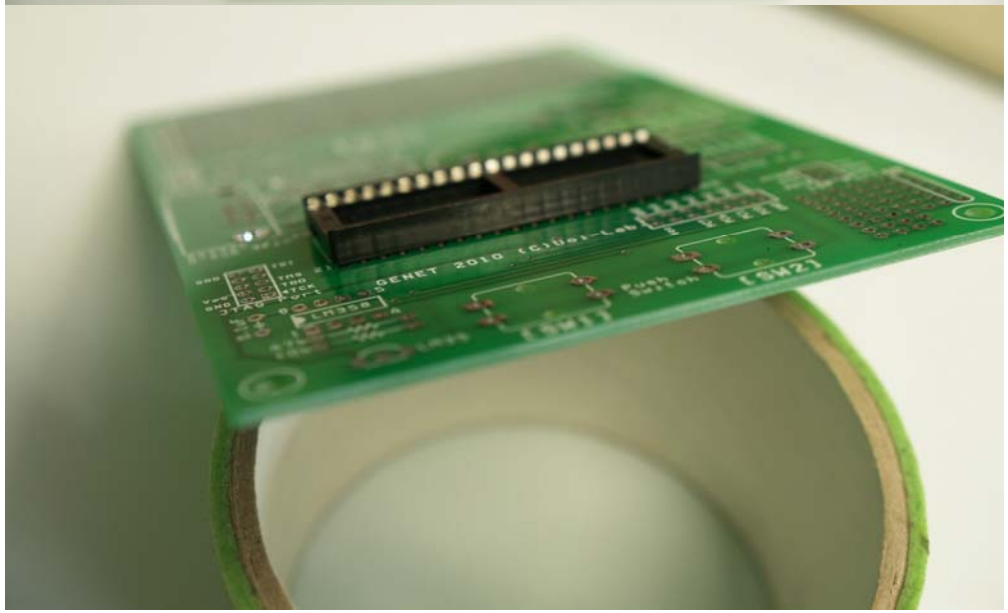
- はんだごては高温になるので取扱いには十分注意してください。席を離れるときは必ず電源を抜いてください。
- 一旦はんだ付けしてしまうと、外すのは大変です。つける前に、場所や向きに誤りがないか十分確かめてください。間違って取り付けてしまった場合は無理に取ろうとせず、アシスタントを呼んでください。
- 電気のショートは非常に怖いです。濡れた手で作業しないでください。また、切り取った裸線やはんだカスはショートの原因になりますので、速やかに隔離しましょう。
- 取り付けたい部品は足を切らずにそのまま基板の穴に通す。向きがあるものは注意！取り付けは背の低い部品から！

4. 実験ボード作成

はんだ付けのポイント

- 接合したい部品と基板の両方の接合部を同時にはんだごての先で手早く温める。こて先の当て方(角度)がポイントです。時間をかけると、端子を通じて電子部品が破損したり、基板が焦げます。また、はんだ自体も煮えて(フラックスが気化してしまう)接合しにくくなります。こて先の掃除もまめに行いましょう。
- はんだごて先端部で接合部を温めながら、はんだをはんだごて先端部に近付け、はんだが溶けだしたら、一気に滑り込ませるように適量を溶かしきり、すぐにはんだごてを離しましょう。山盛りはんだは不良の原因になりやすいです。
- はんだがのったら、こてをそっと離し、冷ますと固まります。こてを離したときに一息吹きかけてあげると、さっと固まります。最後に余分な部品足をニッパで切断します。



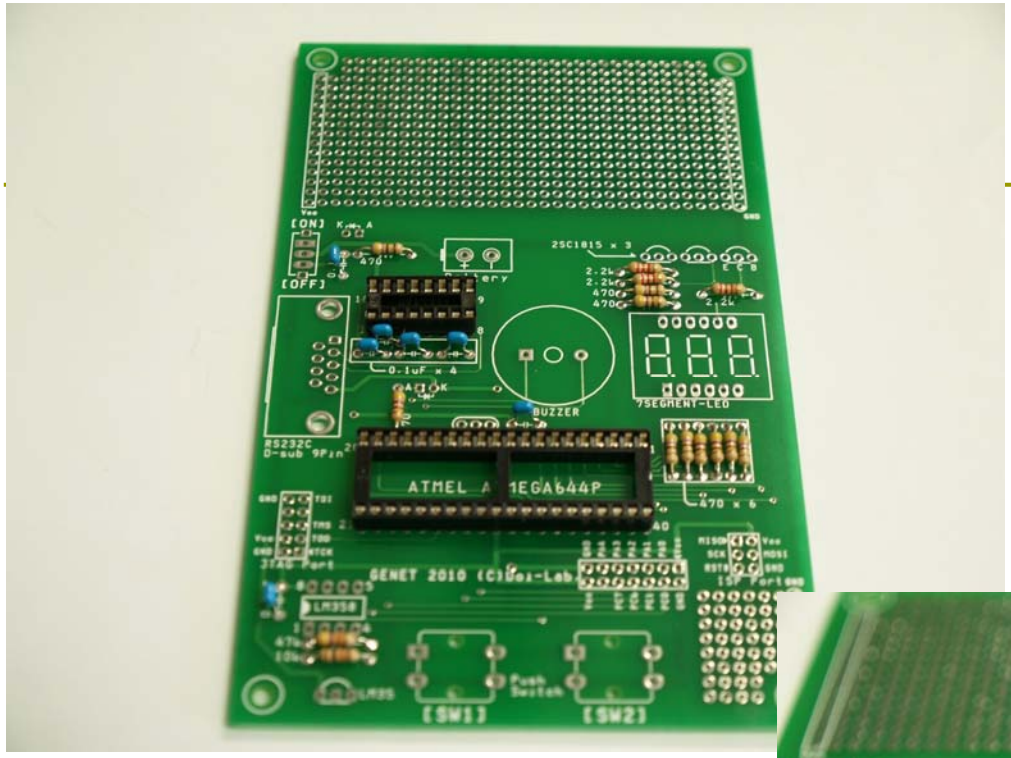


初めにICソケットをハンダ付けします。

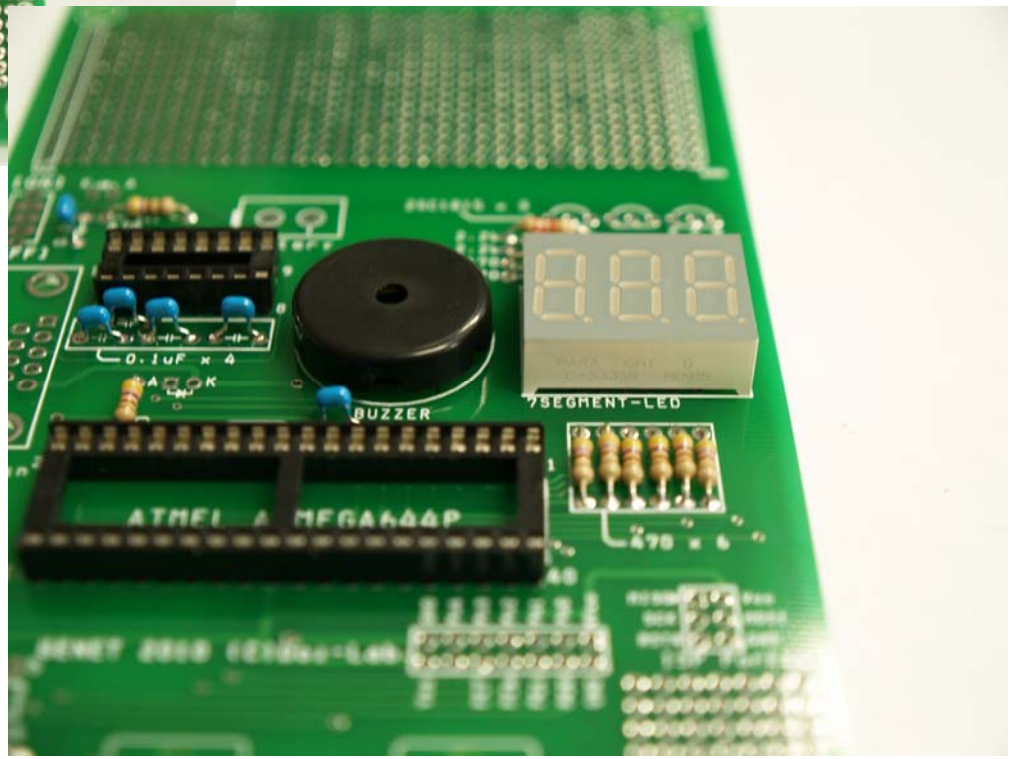
IC保護のスポンジ等を利用して、傾かないように、対角をハンダ付けします。

ICソケットの向き、丸い切りかきに注意します。

傾いていないかを確認のあと、全体をハンダ付けします。

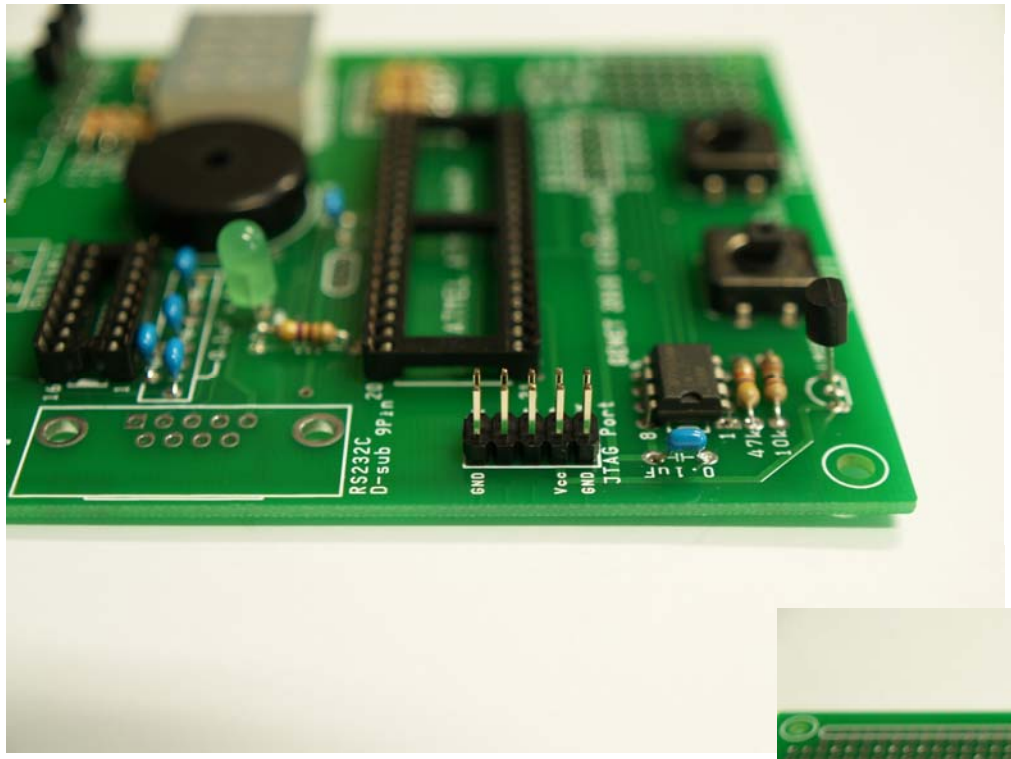


続いて、背の低い部品、
抵抗、コンデンサをハンダ付けします。



さらに、ブザー、7セグLEDをハンダ付けします。

7セグLEDの向き、小数点が下にくるように注意します。



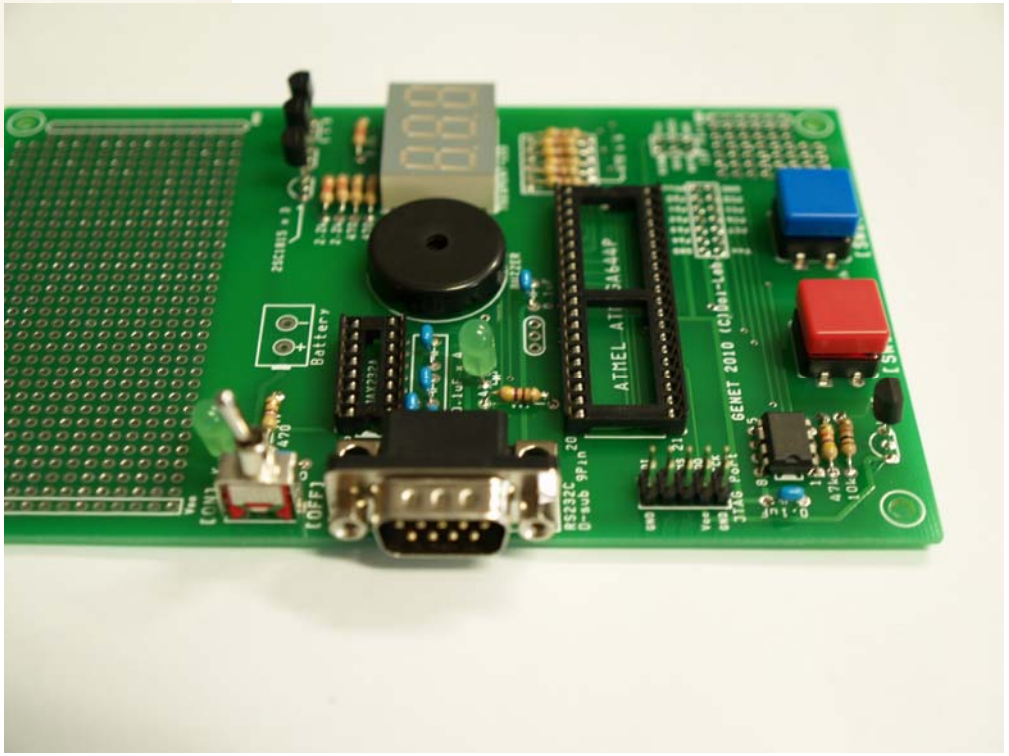
次にピンヘッダをハンダ付けします。

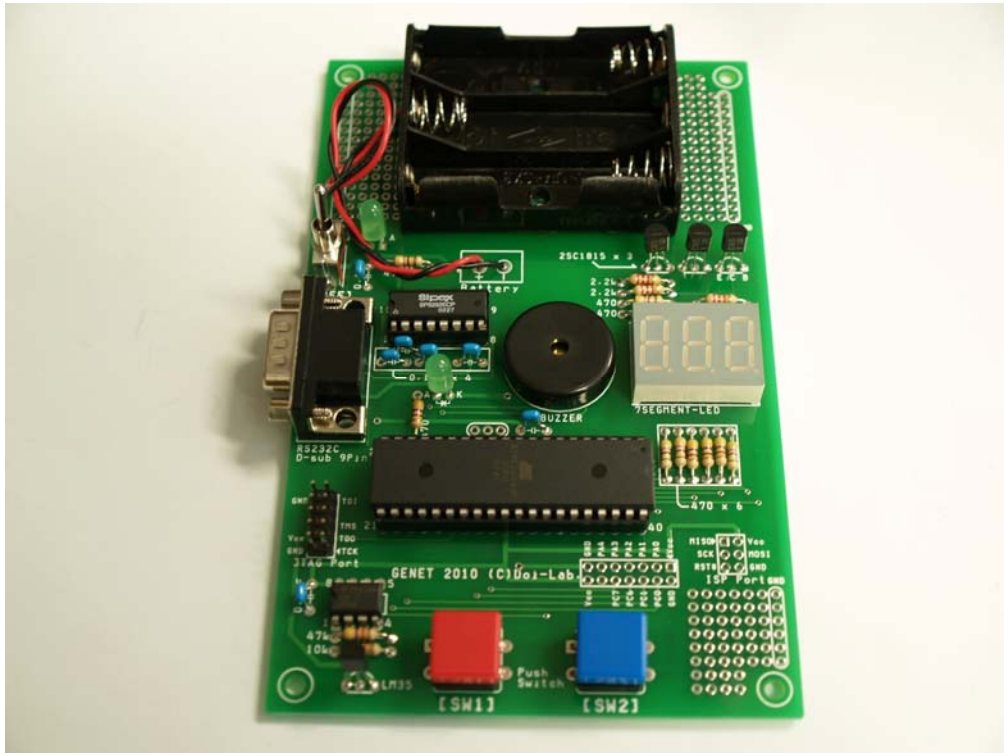
ICソケットと同様に傾かないよう注意し、対角から順にハンダ付けします。

そのあと、オペアンプIC、温度センサ、トランジスタ、LED、と進めます。

温度センサ、トランジスタはリードを切りつめないようにします。

Dサブコネクタ、スイッチ類をハンダ付けします。





電池ボックスを取り付け、
ICソケットにICを取り付け
て完成です。

動作確認

動作確認用プログラム仕様

- 電源投入時に7セグLEDが全点灯を約2秒間維持。その後、3回点滅表示(321のカウントダウン、全消灯を約1秒間隔)後、内部カウント表示へ。7セグLEDの点滅表示に合わせビープ音と赤色LED(ブザー左横)が点滅。
  → 7セグLED 、赤色LED、圧電スピーカの動作を確認
  電源スイッチ脇の赤色LEDは電源ON時は常時点灯
- 右側ボタンにより表示を温度(小数点以下1桁、約1秒間隔更新)表示に切り替え。
  →温度センサ、アンプ、AD変換、右側ボタンの動作を確認。
- 左側ボタンにより表示を内部カウント(プログラム起動時より0～255をループ、約1秒間隔更新)表示に切り替え。
  → 左側ボタンの動作を確認。
- パソコンと接続し、キーボード入力のエコーバックを確認。
  ボーレート2400bps、8ビット、パリティ無し、ストップビット1
  → COMポートの動作を確認。