制御部：命令を実行するための信号を各部へ送る。
データパス部：制御信号に応じて命令の処理を行う。
クロック：各部が同期して動作を行うためのタイミングの基準信号。
ビット長＝1 語（ワード）
マイコンの内部状態：レジスタトランスファーレベルにおける動作の表示。

・フェッチ
MAR←PC；PC の値を MAR に代入。フェッチするアドレスの指定。
PC←PC+1；次にフェッチするアドレスの指定。
PC には次に実行すべき命令の番地が入っている。

・　デコード
IR←MM(MAR)；メインメモリ MM の MAR で指定された番地の内容を命令レジスタに読み込む。
MAR←IR ’4：15’；MAR にオペレンド、アドレス部の値を代入。

・エグゼキュート
R←MM(MAR)：メインメモリの MAR で指定された番地の内容を汎用レジスタに読み込む。

ST：MM(MAR)←R；
LD：R←MM(MAR)；
ADD：R←MM(MAR)+R；
SUB：R←MM(MAR)-R：
B：PC←OPRD; 分岐
BZ：PC←OPRD for Z=1;
BN：PC←OPRD for N=1;

・ マイコンのリブート時（初期状態）
  PC=0；
  MAR←PC , PC←PC+1；
  IR←MM(MAR)； IR=0004H
  制御部←0H (OP) , MAR←0004H(OPRD)
  R←MM(4H)； R=0001H
  ↓
  制御部へ HLT が行くまでサイクル。

・ 主記憶の実現。
  DRAM→コンデンサ
  SRAM→FF（コストが高い、安定的に保存したいときに使う）

- ASC（アドレス 12 ビット：16 進数 3 桁）
  アドレス＝2 の 12 乗個。
  1 アドレスは 16bit=2byte なのでアドレスの大きさは 8K バイト。

- マイコンにとって命令とデータの区別はない。
  PC で指定されると命令
  命令で指定されるとデータである。

- データ、プログラムは実行時に書き換え可能である。
  もともとはハード固定の LSI をつくっていたが汎用性がなく面倒なので
  どんな機種でも使えるマイコンを作った。

- ノイマンボトルネック
  CPU とメモリ間の多量のデータ転送による処理能力低下。
  最近は隙間にキャッシュを置いて高速化している。
  マイコンのように単純なものだとあらわれやすい。

- MAR：メモリアドレスレジスタ
  番地を記憶しておく専用のレジスタ→マイコンが処理を行うための一時的な記憶。

- 機械命令の実現
  機械命令＝制御内容と形式。

- 機械命令の制御
  1. データの転送（ST,LD）
  2. 演算（四則演算、論理演算、左右シフト）
  3. プログラム制御（分岐命令、テスト分岐命令、停止命令）
  4. 入出力
  5. システム制御（特権命令←システムコール機能で呼び出し）

- スタックメモリ
  Push（書き込み）
  Pop（取り出し）
  メモリが下から積み上がる形式で取り出しは必ず上から。
  今、どこまで利用しているかはスタックポインタで管理。

・ オペランド数と命令長
命令長が長いと命令数が少なくなる。
→処理の実行サイクルの繰り返しが少なくなる。
→高速化へつながる。

・ アドレッシング：アドレス指定方法
1. 直接アドレス指定（直接番地指定）
オペランドの内容が処理すべき対象（＝実行番地）
2. 間接アドレス指定（関節番地指定）
オペランドで指定された番地の内容が処理すべき対象
利点：プログラムによって柔軟性を持たせられる
欠点：一手間増えるのでアクセス速度が落ちる
3. イミディエイトアドレス指定（即値番地指定）
オペランドの内容はデータでなく数値データ。
利点：読み出す動作がなくなるので高速化
欠点：データを取り出す範囲はイミディエイトのビット長で制限される
4. インデックスレジスタ（IX）参照番地指定
実行番地＝IX+オペランドで指定されたアドレス
今日は 15 日なので 15 番とあてそこから 10 ずつ足していくような感じ

5. PC 参照番地指定
実行番地＝PC±オペランドの指定
今いるところからどっちかへいく。すごろく。
6. 相対参照番地指定
BR を基点にしてアドレスを指定。プログラム移動可能。
ジャンプ命令の必要がない。命令数が少なくて済む。

・ 固定小数点形式
4 ビットの場合、整数なら+999～+000、-999～-000。
扱える数の範囲が狭い　が有効桁数が多い。
・ 浮動小数点形式
4 ビットの場合、+0.00*10^-4～+0.99*10^5
範囲は広いが、有効桁数が少ない。
ゲタ数=2^(n-1)
+0.92*10^5　→メモリ内では　+992　ゲタが 4 なので。
+092　→　実際の値は+0.92*10^-4