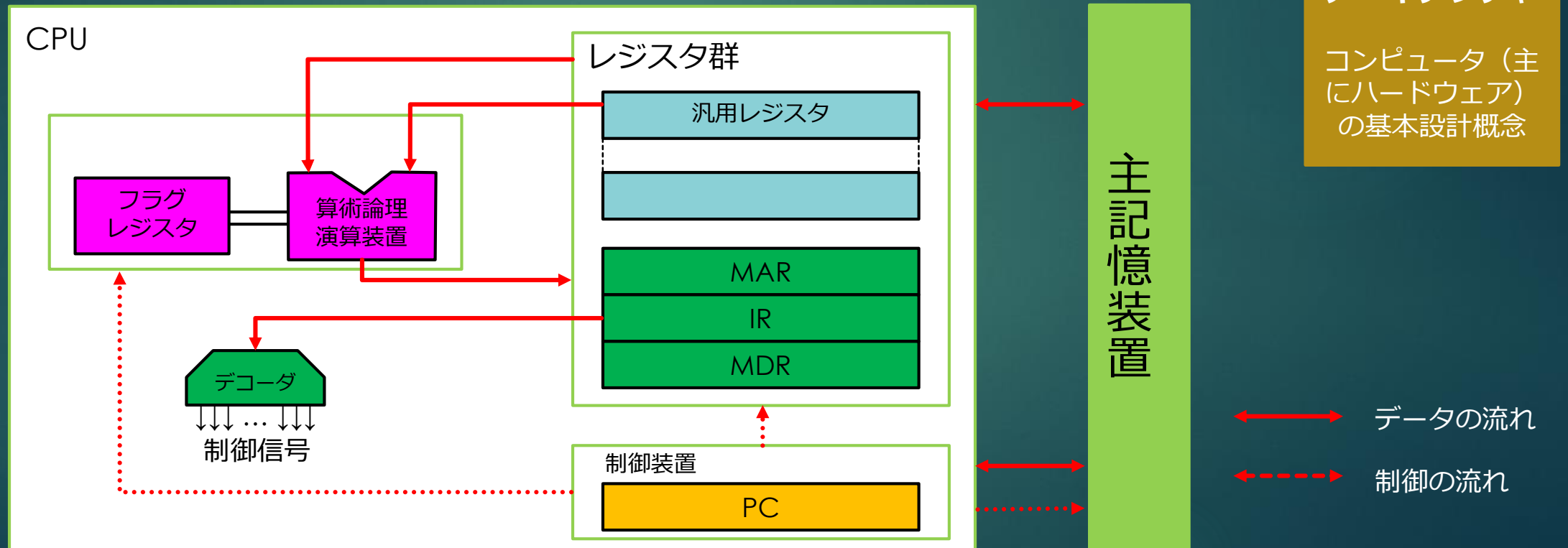


応用情報技術者試験

第3章 ハードウェア

1. 機械語命令の実行 プロセッサの構成要素

プロセッサ(CPU)は、コンピュータにおいて制御と演算を担当する中核部分である。その構成要素は**アーキテクチャ**（設計）によって異なる。



プロセッサの構成要素

名称	機能
ALU (Arithmetic Logic Unit) 算術論理演算装置	算術演算や論理演算などの演算を行う
PC (Program Counter) プログラムカウンタ /命令アドレスレジスタ	現在実行中の（もしくは次に実行する）命令の格納場所（アドレス）を格納する
GR (General purpose Register) 汎用レジスタ	演算対象やアドレス情報といった様々な情報を格納する
FR (Flag Register) フラグレジスタ	演算結果の正負に関する情報を格納する
IR (Instruction Register) 命令レジスタ	命令そのものを格納する
ID (Instruction Decoder) デコーダ	命令を解釈し、命令に応じた制御信号を生成する
MAR (Memory Address Register) メモリアドレスレジスタ	アクセス対象となるメモリの位置（番地）情報を格納する
MDR (Memory Data Register) メモリデータレジスタ	メモリから読み込むデータまたはメモリに書き込むデータの内容を格納する

プロセッサの構成要素

- ▶ **プログラムカウンタ(PC)**と**命令レジスタ(IR)**は密接に関係している。プログラムカウンタがさすメモリに格納された命令は命令レジスタに取り込まれる。
- ▶ 同様に**メモリアドレスレジスタ(MAR)**と**メモリデータレジスタ(MDR)**も密接に関係している。メモリアドレスレジスタのさすメモリに格納されたデータが、メモリデータレジスタに取り込まれる。
- ▶ 命令の取り出しを**命令フェッチ**、処理対象となるデータの取り出しを**オペランドフェッチ**と呼ぶ。

命令の構成

▶ 一般に一つの命令は数バイトで構成され、各命令の長さ（命令語長）はプロセッサのアーキテクチャによって異なる。

- ・ **命令コード部**：命令の内容を表す
- ・ **オペランド部**：レジスタやメモリアドレスなど、演算対象の情報を表す

オペランド

演算対象となる値
や変数

32ビット(4バイト)命令の例:



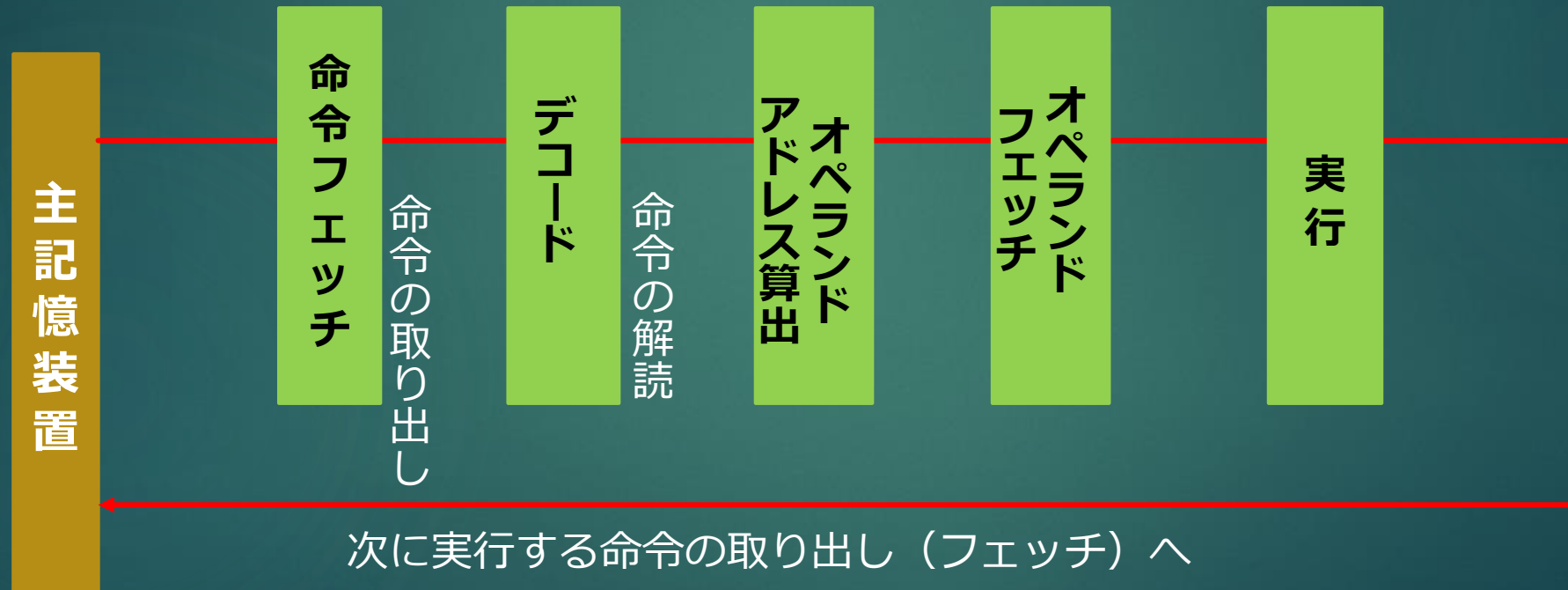
<レジスタ部>
処理対象となるレジスタを指定する

<アドレス修飾部>
アドレス修飾があればその種別を指定する

<アドレス定数部>
処理対象データの格納されたアドレスを指定する

命令の実行過程

- ▶ 主記憶に格納された命令やデータは、次の過程でプロセッサに取り込まれ、実行される。



オペランドアドレス算出

- ▶ 機械語命令のアドレス定数部で指定されたアドレスにデータが格納されているという保証はない。それはアドレスが**修飾**されている可能性があるため。
- ▶ そのような修飾を解決し、**データを格納する真のアドレス（有効アドレス）を求める段階**が命令実行過程におけるオペランド算出である。
- ▶ **（アドレス）修飾**とは有効アドレスを求めるため、アドレス定数部の値に何らかの操作が必要であることを意味する。

アドレス修飾の方法

名称		求められる有効アドレス
絶対	直接アドレス指定方式	アドレスの定数部の値
	間接アドレス指定方式	(アドレス定数部の値) 番地に格納された値
相対	自己相対アドレス指定方式	アドレス定数部の値 + PCの値
	ベースアドレス指定方式 (ディスプレイースメント)	アドレス定数部の値 + ベースレジスタの値
	インデックスアドレス指定方式	アドレス定数部の値 + インデックスレジスタの値

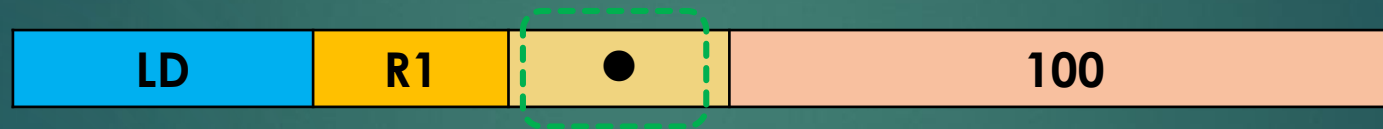
相対アドレス指定とリロケータブル(参考)

- ▶ アドレス定数部の値を100としたとき、自己相対アドレス指定は「今実行している命令（PCの値）から100番地だけ離れたメモリ」を、ベースアドレス指定は「プログラムの先頭（ベースレジスタの値）から100番地だけ離れたメモリ」を指定する方式である。このような面倒なアドレス指定をする理由は、プログラムの再配置性にある。
- ▶ プログラムは、制御上の理由で配置された位置がずれる場合がある。そのような場合であっても、自己相対アドレス指定やベースアドレス指定を行っている限り、プログラムを修正することなく実行できる。理由として、プログラムの配置換えに伴い、アドレス指定の基準となるPCやベースレジスタの値も変化するため、位置にかかわらず正しいメモリをさすことができるからである。
- ▶ このような位置に関するプログラムの性質が、**リロケータブル（再配置可能性）**である。リロケータブルは、仮想記憶が一般化する以前は、プログラムに必須の性質であった。

オペランドアドレス算出

主記憶

20		ベースレジスタ		5		インデックスレジスタ		50		プログラムカウンタ	
番地		100		105		110		120		150	
...		110		...		200		...		250	
...			100		...	
...		120		



アドレス修飾の方式
(複数指定化)

<直接アドレス指定の場合>

有効アドレス=100なので
100番地の内容(110)をレジスタR1にロードする

<インデックスアドレス指定の場合>

有効アドレス=100+5=105なので
105番地の内容(200)をレジスタR1にロードする

<ベースアドレス指定の場合>

有効アドレス=100+20=120なので
120番地の内容(100)をレジスタR1にロードする

<間接アドレス指定の場合>

有効アドレス=100番地の内容=110なので
110番地の内容(250)をレジスタR1にロードする

<自己相対アドレス指定の場合>

有効アドレス=100+50=150なので
150番地の内容(120)をレジスタR1にロードする

オペランドフェッチ

- ▶ メモリの有効アドレスに格納されたデータを、プロセッサ（内のレジスタ）に取り込む段階を**オペランドフェッチ**という。データが複数バイトに及ぶ場合は、その並び順（**エンディアン**）に従って取り込まれることになる。
- ▶ エンディアン方式

ビッグエンディアン

データの**上位バイトから順**にメモリに並べる方式

リトルエンディアン

データの**下位バイトから順**にメモリに並べる方式

ビッグエンディアン /リトルエンディアン

<例>

4バイトのデータを
32ビットのレジスタ
に取り込む

上位バイトから
並べる

下位バイトから
並べる

ビッグエンディアン

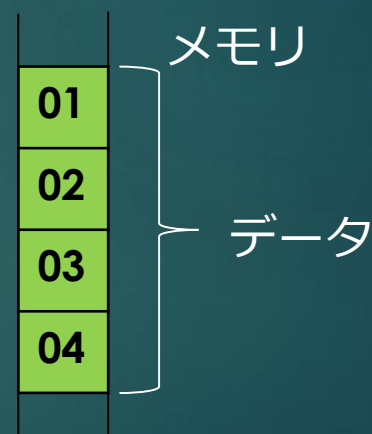


オペランド
フェッチ



リトルエンディアン

データレジスタ



データレジスタ

2. プロセッサアーキテクチャ

マイクロプログラム方式とワイヤードロジック方式

- ▶ 命令の実行に際して、制御装置は実行に必要な構成要素に信号を送り、制御する。それをどのように実現しているかについて、プロセッサはマイクロプログラム方式とワイヤードロジック方式に大別できる。

ワイヤードロジック方式	必要な制御を、各命令コードごとにすべて ハードウェア で実現する方式
マイクロプログラム方式	各命令は、より単純な命令を用いた マイクロプログラムで表される 。マイクロプログラムの命令自体は、ハードウェアで実現する

- ▶ これらの方式を、**命令の実行速度**で比較するとワイヤードロジックの方が高速
- ▶ 一方で**命令の追加・変更**については、マイクロプログラム方式が柔軟に対応可能

RISCとCISC

- ▶ 各プロセッサが実行可能な命令の集まりを**命令セット**と呼ぶ。その命令セットの単純さで分類すると、プロセッサはRISCとCISCに分類できる。

RISC	使用頻度の高い 最小限度の命令 だけで命令セットを構成する方式
CISC	複雑で 高機能な命令 を用意する方式

RISCとCISC

- ▶ RISCとCISCの差異は次のようになる。

	RISC	CISC
命令の種類	少ない	多い
1命令の実行速度	速い	遅い
固定長命令の実現	容易	困難
プログラムのステップ数	多い	少ない
制御	ワイヤードロジック	マイクロプログラム

3. プロセッサの性能評価 クロック周波数とCPI

- ▶ プロセッサの命令実行は、一定間隔で発生する信号に同期して行われる。このときの同期信号を**クロック**という。また発生頻度を**クロック周波数**とよぶ。
- ▶ 例：クロック周波数が800MHz(メガヘルツ) → 1秒間に 800×10^6 回クロックが発生
- ▶ 同一のアーキテクチャにおいて、**クロック周波数の高いプロセッサほど性能が高い。**

クロック周波数とCPI



- ▶ クロック周波数の逆数を求めれば、1クロックあたりの時間(**クロックサイクル**)を求めることができる。クロック周波数が800Mhzのときの1クロックあたりの時間は、

$$\begin{aligned}\frac{1}{800 \times 10^6} &= 1.25 \times 10^{-9} \text{秒} \\ &= 1.25 \text{ナノ秒}\end{aligned}$$

クロック周波数とCPI

- ▶ **CPI(Cycles Per Instruction)**は、1命令を実行するために必要なクロック数を表す。**CPI=5は、1命令を5サイクルで実行できる**ことを表す。
- ▶ クロック周波数をCPIをもとに、1命令あたりの実行時間を知ることができる。クロック周波数が800MHzでCPIの平均が5であるプロセッサがあるとき、このプロセッサにおける1命令あたりの平均実行時間は、つぎのようになる。

$$1.25 \times 5 = 6.25 \text{ ナノ秒}$$

命令ミックス

- ▶ 実際のプロセッサでは、1命令の実行に必要なクロックサイクル数(CPI)は**命令ごとに異なる**。
- ▶ そのため、CPIの平均値は、**命令ごとの出現頻度を基にした期待値**で計算しなければならない。
- ▶ このときに用いる「命令の出現頻度表」が**命令ミックス**という。

命令ミックス

- ▶ 例：800MHzのプロセッサの場合

命令の種類	所要クロックサイクル数	出現確率
浮動小数点数演算	6	0.2
メモリアクセス	3	0.4
分岐その他	2	0.4

このときCPIの平均値は、

$$\begin{aligned} & \sum(\text{各命令の所要クロックサイクル数} \times \text{出現確率}) \\ &= 6 \times 0.2 + 3 \times 0.4 + 2 \times 0.4 \\ &= 1.2 + 1.2 + 0.8 \\ &= 3.2 \end{aligned}$$

となり、平均命令実行時間は

$$\text{平均命令実行時間} = \frac{1}{800 \times 10^6} \times 3.2[\text{秒}] = 4[\text{ナノ秒}]$$

と計算できる

MIPSとFLOPS

- ▶ プロセッサの性能評価指標として、**MIPS**や**FLOPS**などが用いられる。

MIPS	Million Instructions Per Second 1秒間に実行できる命令の平均値 を、 100万(= 10^6)を単位として表したもの
FLOPS	1秒間に実行できる浮動小数点数の演算の回数 。主に科学技術計算分野で用いる。

MIPSとFLOPS

- ▶ クロック周波数が800MHzで、CPIの平均値が5のプロセッサがあった場合、MIPS値は次のようになる

1秒間の命令実行数/ 10^6

$$= \frac{\left(\frac{\text{クロック周波数}}{\text{CPIの平均値}} \right)}{10^6}$$

$$= \frac{\frac{800 \times 10^6}{5}}{10^6}$$

$$= 160$$

4. プロセッサの高速化 命令パイプライン

- ▶ 命令パイプラインとは、**命令を並列に実行させる**ことで高速化を実現する技術である。具体的に、命令をいくつかの**ステージ**（段階）にわけ、各ステージをオーバーラップさせながら並列に実行させる。このとき、1ステージを処理するために必要な時間を**ピッチ**、同時に実行できる命令数をパイプラインの**深さ**ということもある。

ピッチ

1ステージの実行
時間

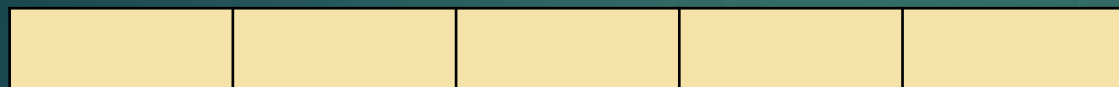
深さ

パイプラインの重
なり数
→同時に実行でき
る命令数

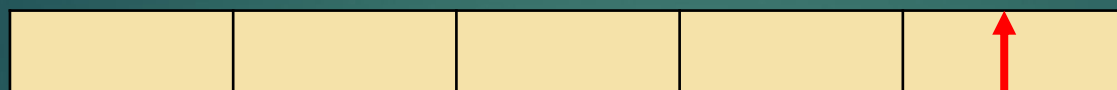
命令パイプライン

時間

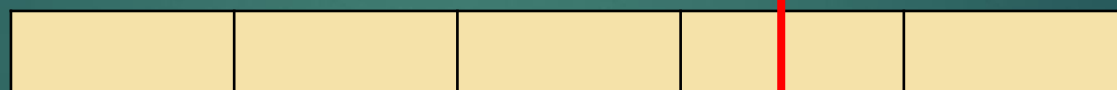
命令i



命令i+1



命令i+2



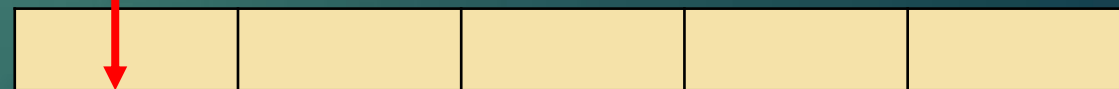
命令i+3



命令i+4



命令i+5



深さ
並列して実行可能
な命令数

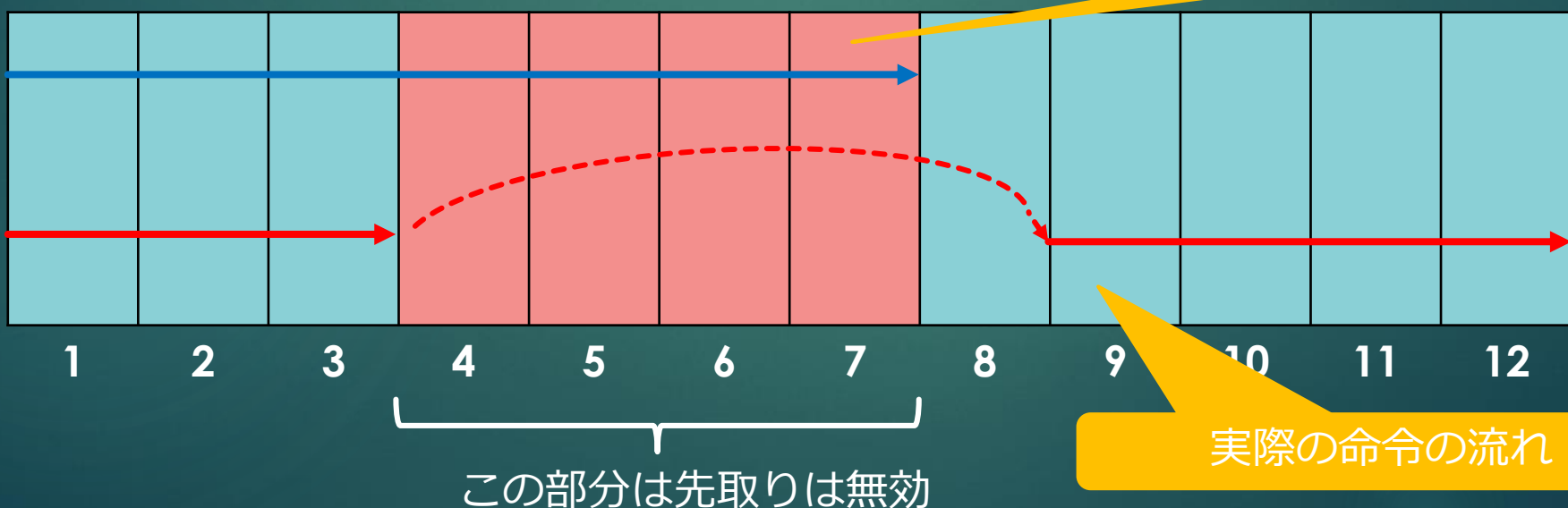
ピッチ

命令パイプライン

- ▶ 1命令が5ステージに分割されて実行される場合を考える（先のスライド）。パイプラインを用いない場合は n この命令実行には $(5 \times n)$ サイクルが必要になる。しかし、先のスライドのような命令パイプラインを使用すると**5命令が並列に実行される**ことになり、サイクル数は **$1/n(=n \text{ サイクル})$ に減少する**。ただし、5命令が重なるまで4ステージ分のオーバーヘッドが生じるため、厳密には $(n+4)$ サイクルになる

ハザード

- ▶ 命令パイプラインは次に実行する命令を「先取り」することによって高速化を実現する。そのため、分岐命令の実行により先取りした命令が無効になったり、メモリの競合による衝突が生じたりすることでパイプラインが乱れることがある。このような乱れを**ハザード**とよぶ。



ハザード

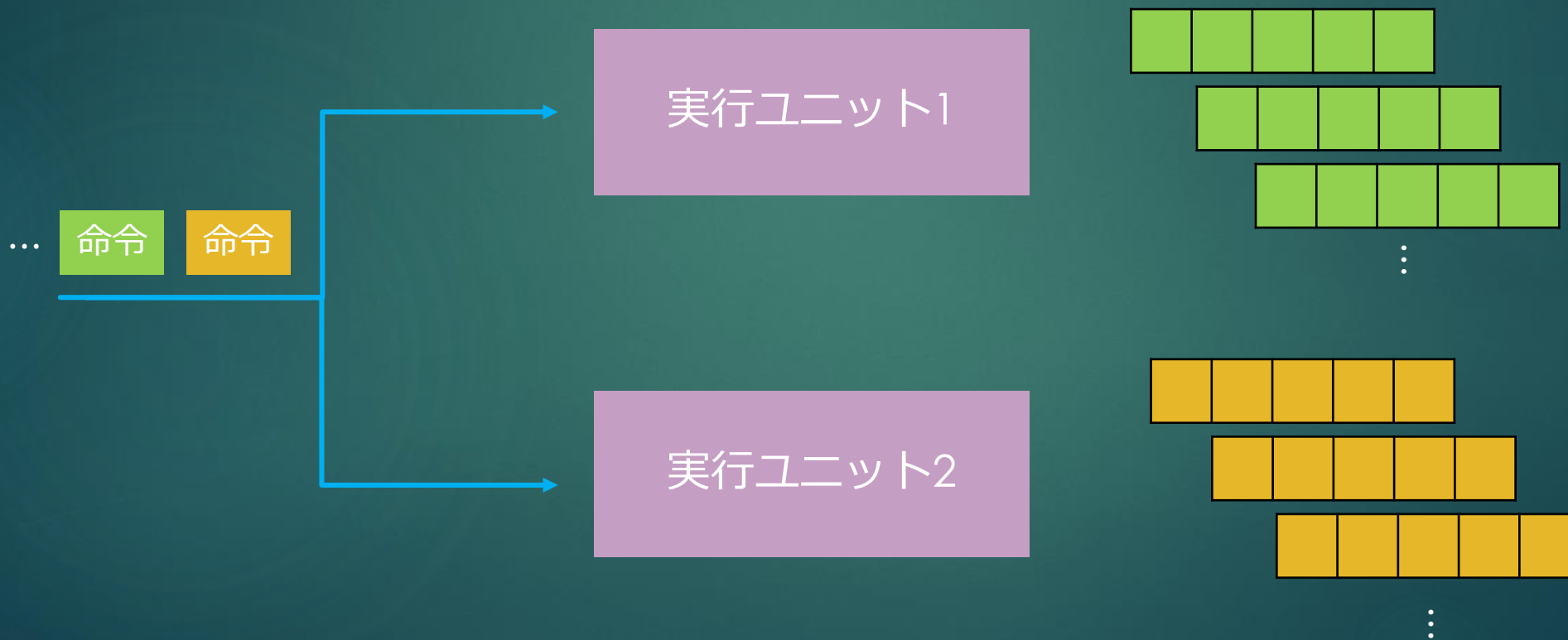
- ▶ 先の図は分岐命令でのハザード（**制御ハザード**）を示している。プロセッサが命令3を実行しているとき、パイプラインは命令7まで先取りしている。ここで命令3の内容が「命令9にジャンプする」ものであったとき、先取した命令4~7は破棄しなければならない。パイプラインはいったんリセットされ、命令9からやりなおす。
- ▶ ハザードを抑制できれば、パイプラインが有効に機能する
 - 分岐命令を少なくするようプログラムする
 - 分岐先を予測し、分岐先の命令を先取りする（投機実行）

スーパパイプライン

- ▶ スーパパイプラインは命令パイプラインを発展させた高速化技法で、命令の**ステージをより細分化すること**でパイプラインの深さを増し、同時に実行できる命令数を上げる技術である
- ▶ スーパパイプラインはパイプラインの並列度を増加させる。

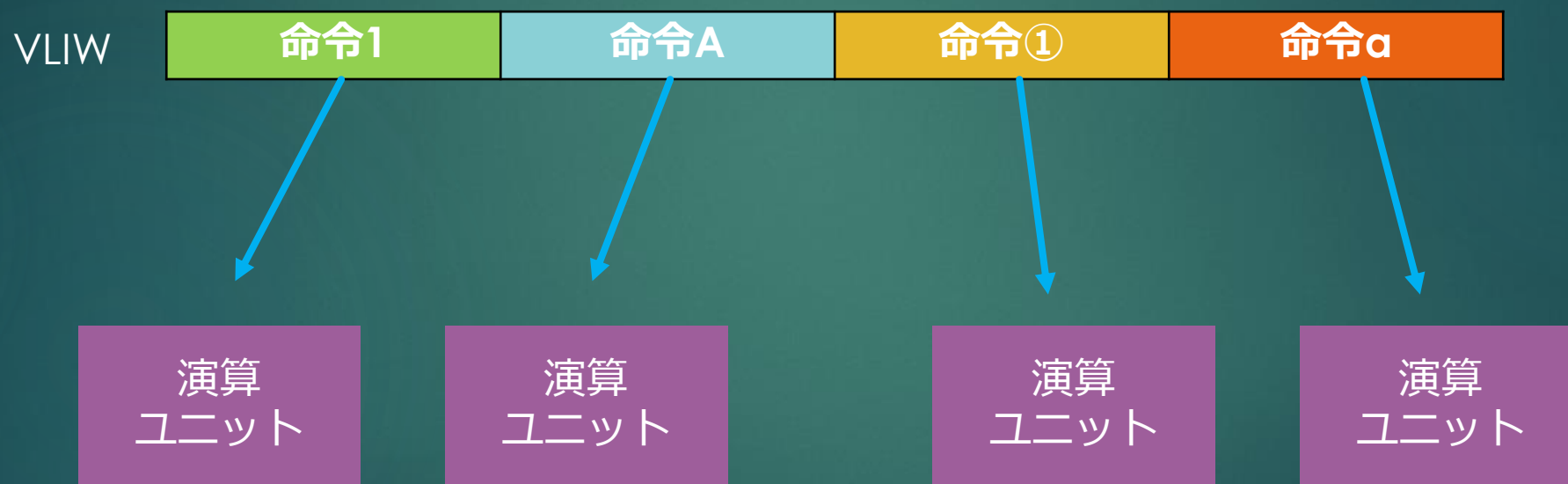
スーパースカラ

- ▶ **スーパースカラ**はプロセッサ内に命令の実行ユニットを複数用意し、各ユニットで命令パイプラインを実行させる技術である。**命令パイプラインそのものが多重化**されることになる。



VLIW (Very Long Instruction Word)

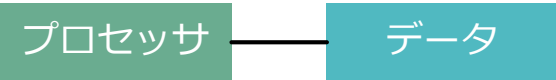
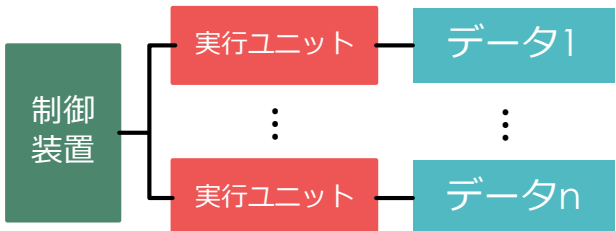
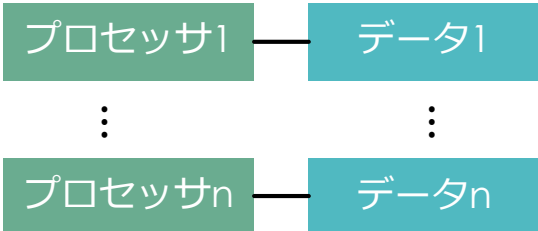
- ▶ 同時に実行可能な複数の命令をまとめて一つの長形式命令(VLIW)を作成し、これを複数の演算ユニットで同時に実行することで高速化を実現する方式である。



マルチプロセッサ

- ▶ **マルチプロセッサ**は、プロセッサそのものを複数用意し、それらを並列に動作させる高速化技術である。プロセッサ単体の高速化技法ではないが、全体として性能が引き上げられる。

マルチプロセッサ

名称	命令の流れ	データの流れ	構成イメージ
SISD (Single Instruction stream / Single Data stream)	単一	単一	
SIMD (Single Instruction Stream / Multiple Data stream)	単一	複数	
MIMD (Multiple Instruction stream / Multiple Data stream)	複数	複数	

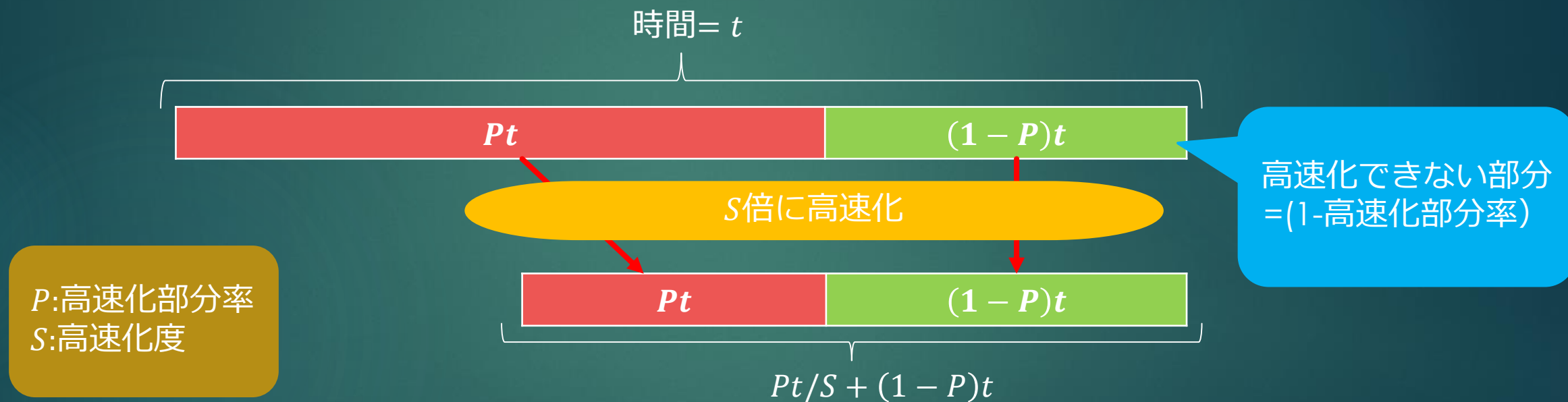
スーパーコンピュータに用いられた**アレイプロセッサ**は**SIMD**に該当する。一般的なマルチプロセッサは**MIMD**に該当する。

マルチコアプロセッサ

- ▶ **マルチコアプロセッサ**は一つのプロセッサ内に演算を行う**コア**を複数持ち、**コアを並列に動作させる**ことで高速化を実現する方式である。外見上は一つのプロセッサとして認識され、消費電力や発熱を抑えながら処理性能を向上させるように設計されている。

Amdahl(アムダール)の法則

- ▶ プロセッサがいくら高速化されてもそのほかが高速化されなければ、高速化には限度がある。その関係を示すモデルがアムダールの法則である。



$$\text{速度向上比} = \frac{t}{Pt/S + (1-P)t} = \frac{1}{P/S(1-P)}$$

Amdahl(アムダール)の法則

- ▶ たとえば、全体80%を5倍に高速化できる場合、速度向上比は

$$\frac{1}{\frac{0.8}{5} + 0.2} \doteq 2.8[\text{倍}]$$

となる。なお、高速化部分率が0.8であれば、高速化度(S)をどれだけ高めたとしても、速度向上比は $(1/0.2)=5$ に近づくだけで、これを超えることはない。高速化には**高速化部分率による限界**がある。

5. メモリアーキテクチャ 集積回路の種別

- ▶ メモリは集積回路(IC)で構築される。

バイポーラ型	バイポーラ（双極）トランジスタを用いる
CMOS型	PMOSとNIMOSの2種類のMOSトランジスタを組み合わせで用いる
BiCMOS型	CMOS型とバイポーラトランジスタを組み合わせで用いる

バイポーラ型とCMOS型の比較

	バイポーラ型	CMOS型
動作速度	速い	遅い
電流出力	大きい	小さい
集積度	低い	高い
消費電力	大きい	小さい
電源ノイズへの耐性	弱い	強い
静電気への耐性	強い	弱い
製造コスト	高い	低い

BiCMOS型は上の二つの特徴を組み合わせ、高速・高集積度を実現している

RAMの分類

RAMは、DRAMとSRAMに分類することができる。

- ▶ **RAM(Random Access Memory)**

主記憶の作業域を構築するメモリ。電源断で記憶内容を失う（揮発性）

- ▶ **DRAM(Dynamic RAM)**

RAMの一種で主記憶に用いる。内容の保持のため**リフレッシュ動作が必要**

- ▶ **SRAM(Static RAM)**

RAMの一種でキャッシュメモリに用いる。フリップフロップ回路で内容保持するため、**リフレッシュ動作が不要**

RAMの分類

	DRAM	SRAM
メモリセルの構成	コンデンサと MOSトランジスタ	フリップフロップ
アクセス速度	遅い	速い
集積度	高い	低い
主な用途	主記憶	レジスタ/ キャッシュメモリ
製造コスト	低い	高い

- ▶ DRAMは内部構成が単純なのでビット当たりの面積が小さく、高集積化を実現することができる。そのため、ビットあたりの単価も安価であるが、アクセス速度はSRAMのほうが速い。

ROMの分類

▶ ROM(Read Only Memory)

読み出し専用に使われるメモリ。電源断でも記憶内容を保持する（不揮発性）

名称		空き部分への追記	既存部分の消去 / 上書き	消去手段
マスクROM		×	×	
ユーザ プログラマブル ROM	PROM	○	×	
	EPROM	○	○	紫外線
	EEPROM (フラッシュメモリ)	○	○	高電圧

誤り制御

- ▶ メモリの誤り制御には**ハミング符号**を用いる。
- ▶ これを実装するメモリを**ECC(Error Check and Correct)メモリ**とよぶ。
- ▶ ECCメモリは、64ビットにつき8ビットのハミング符号を付与する。
- ▶ ECCメモリはサーバなど影響の大きいシステムに用いる。

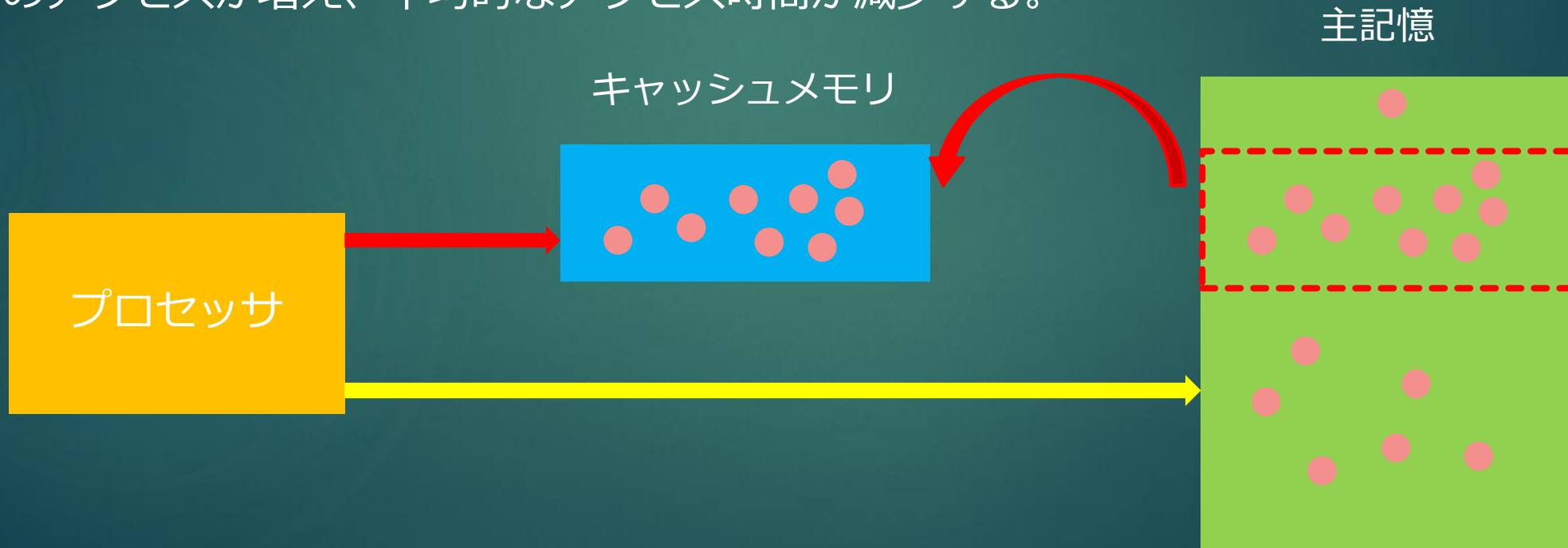
6.メモリの高速化

キャッシュメモリ

▶ キャッシュメモリ

主記憶とプロセッサの間に配置される高速小容量のメモリ

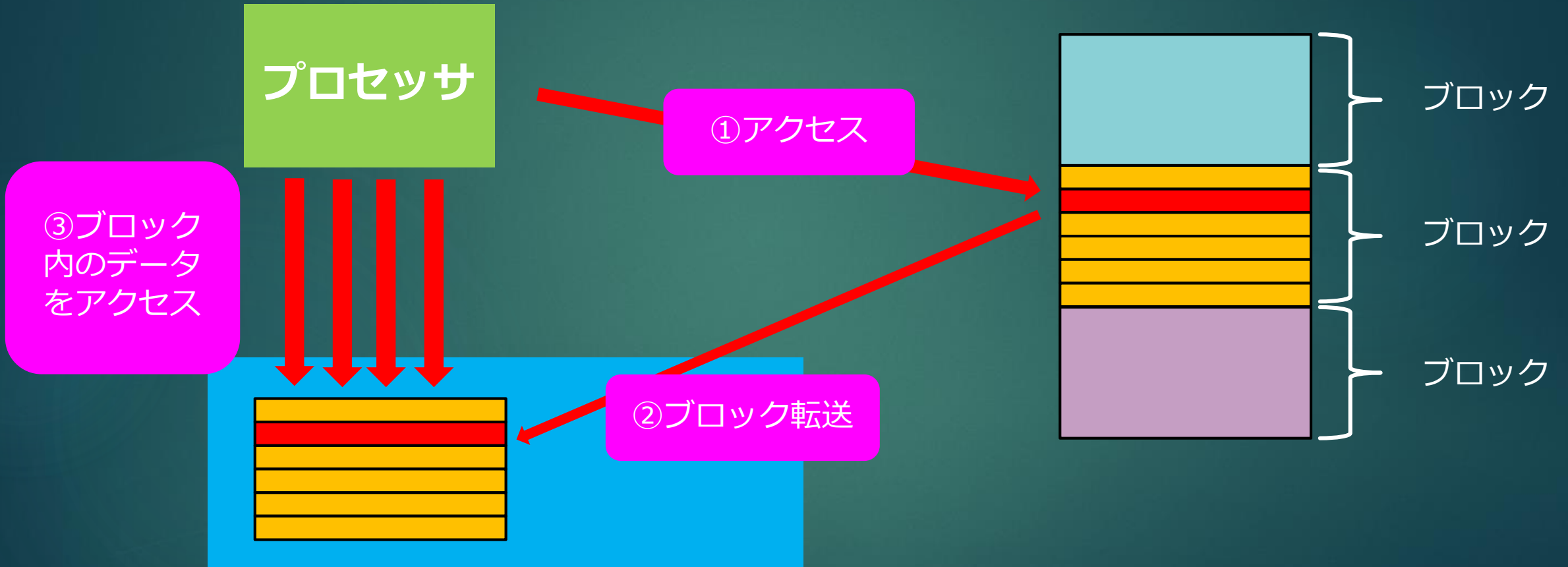
利用頻度の高いプログラムやデータを配置すれば高速なキャッシュメモリへのアクセスが増え、平均的なアクセス時間が減少する。



プログラムの局所参照性

- ▶ **局所参照性**とはある時間間隔で見たとき、プログラムのごく一部の**繰り返し参照する**という性質である。
- ▶ キャッシュメモリはこの局所参照性の性質を用いることでメモリの高速化を十分に発揮している。
- ▶ プログラムの構成によって局所参照性は大きく変わり、分岐やサブルーチン呼び出しが多いと局所参照性は低下する（広範囲を参照するため）

キャッシュメモリの動作



キャッシュメモリの動作

- ▶ 主記憶およびキャッシュメモリは**ブロック単位**で管理されている。
- ▶ プロセッサが主記憶上にあるアドレスをアクセスしたときそのアドレスをアクセスしたとき、**そのアドレスを含むブロックをまとめて**キャッシュメモリに転送する。（局所参照性）
- ▶ キャッシュメモリに**ヒット**する間はプロセッサは高速にキャッシュメモリにアクセスし続ける。ヒットしなければアクセスとブロック転送を行う。
- ▶ キャッシュメモリに空きがなければいずれかのブロックを主記憶に書き戻して空きブロックを確保する。

実効的なアクセス時間

- ▶ 主記憶装置へのアクセス時間を T_m , キャッシュメモリのアクセス時間を T_c , キャッシュメモリのヒット率を r とすると実効アクセス時間（メモリへの平均アクセス時間）

$$\text{実効アクセス時間} = (1 - r) \times T_m + r \times T_c$$

ヒットしない確率は**NFP**という

<例> 主記憶装置へのアクセス時間が100ナノ秒、キャッシュメモリへのアクセス時間が20ナノ秒、キャッシュメモリのヒット率が90%であった場合の平均（実効）アクセス時間は

$$(1 - 0.9) \times 100 + 0.9 \times 20 = 10 + 18 = 28 \text{ [ナノ秒]}$$

キャッシュの制御方法

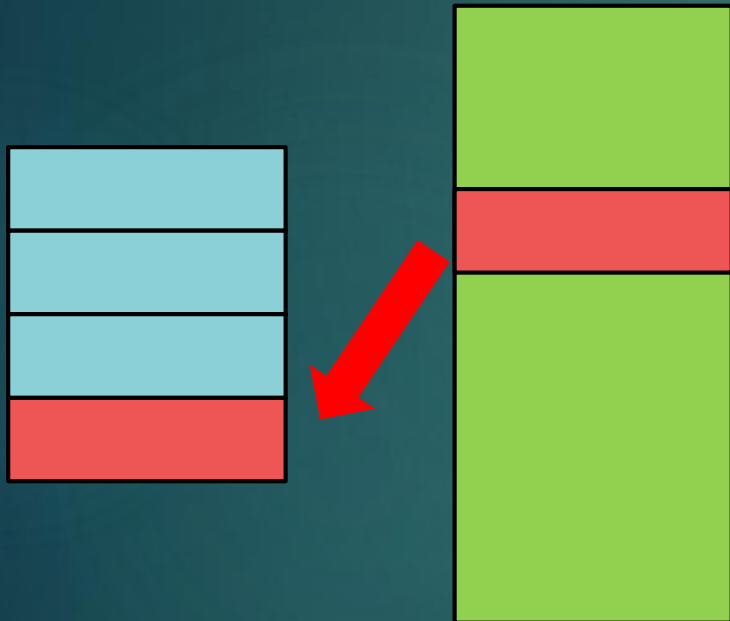
- ▶ キャッシュメモリに行った更新は、いずれかのタイミングで主記憶に反映させなければならない。

ライトスルー方式	キャッシュメモリの更新と同時に、主記憶の該当データも更新する
ライトバック方式	通常はキャッシュメモリのみ更新し、 ブロックが追い出されるタイミング で主記憶に反映する

- ▶ ライトバックは主記憶への書き込み頻度が減少するため、書き込み時にも**キャッシュの効果**が得られる。
- ▶ しかしライトバックは更新と反映の間にタイムラグが生じる。そのため、両者の**コヒーレンシ（同一性）**を保つ特別な制御が必要となる

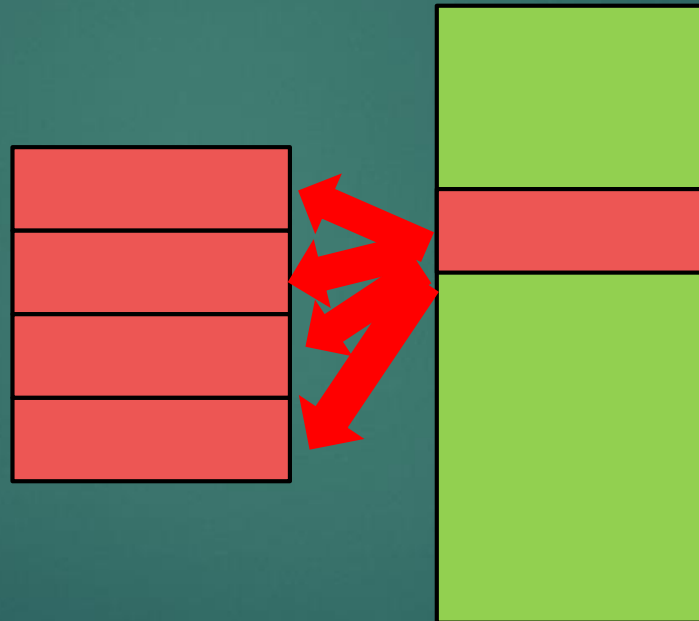
キャッシュの割付け方式

ダイレクトマッピング



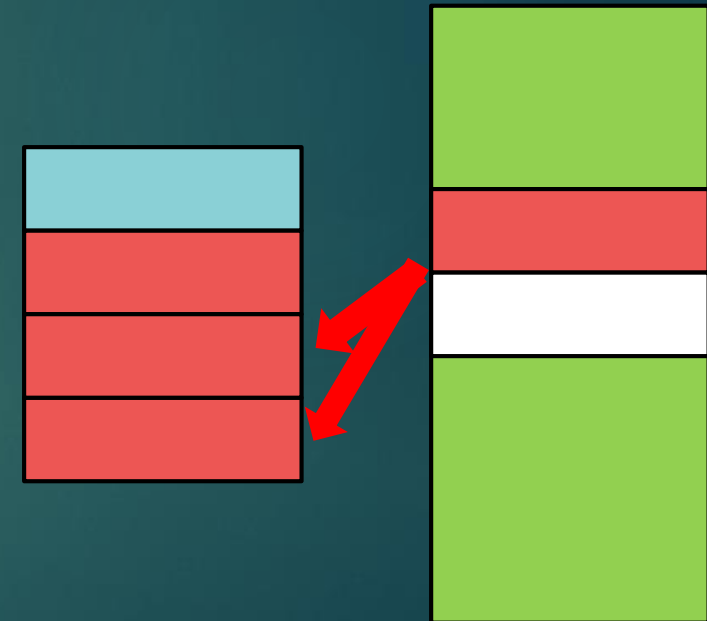
ブロック番号から一意に求まる

フルアソシエティブ



任意のロケーションに配置
ブロック番号とロケーションとの対応は、連想メモリで管理

セットアソシエティブ

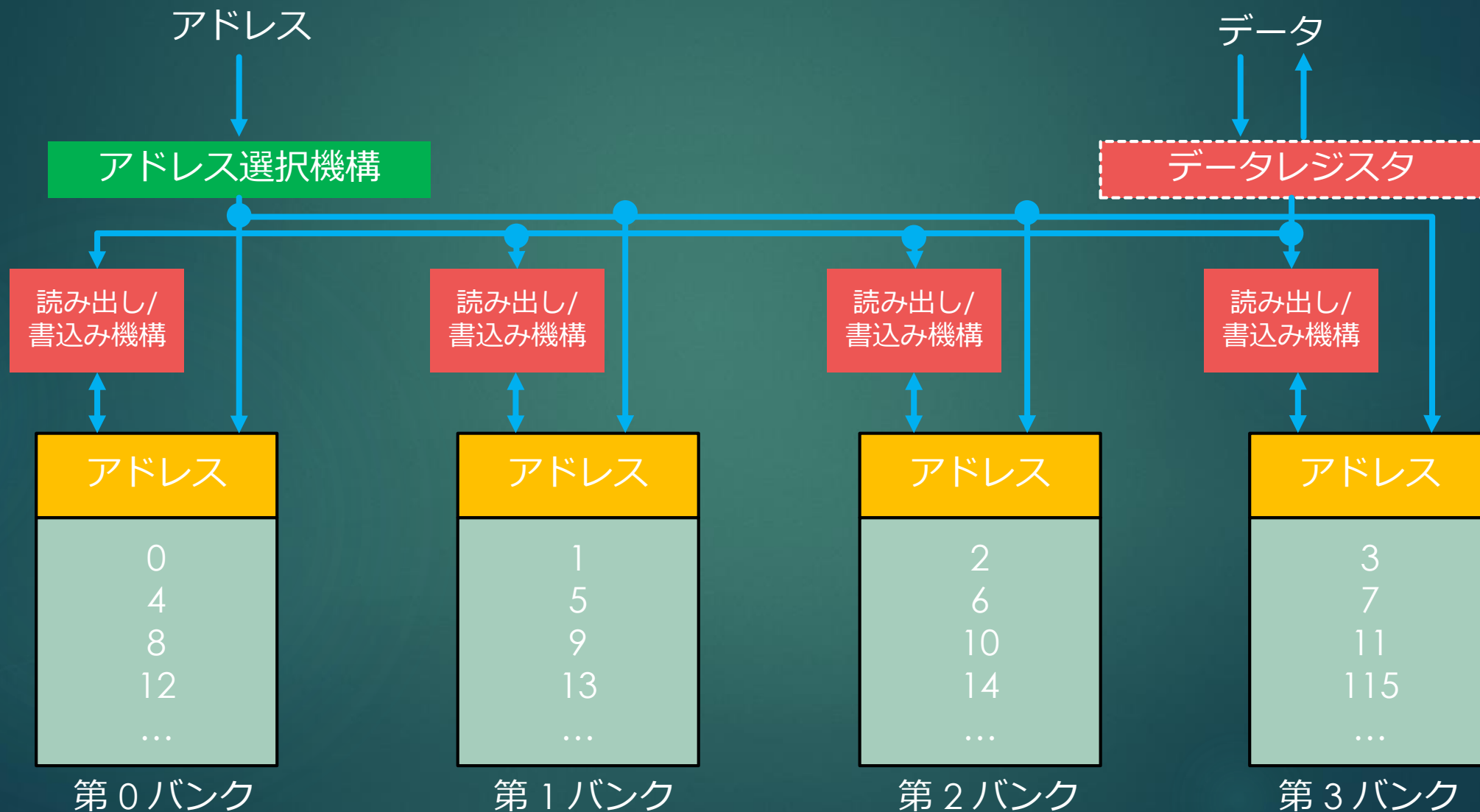


ブロックをいくつかまとめたセットで管理
セットのロケーションは一意だが、その中であればどこでも配置可

メモリアンタリーブ

- ▶ **メモリアンタリーブ**はメモリを複数の独立した“**バンク**”と呼ばれる区画に分け、専用の制御機構によって、**各バンクに対して並列にアクセスすることによって高速化を行う方式**である。

メモリアンタリーブ



メモリーインタリーブ

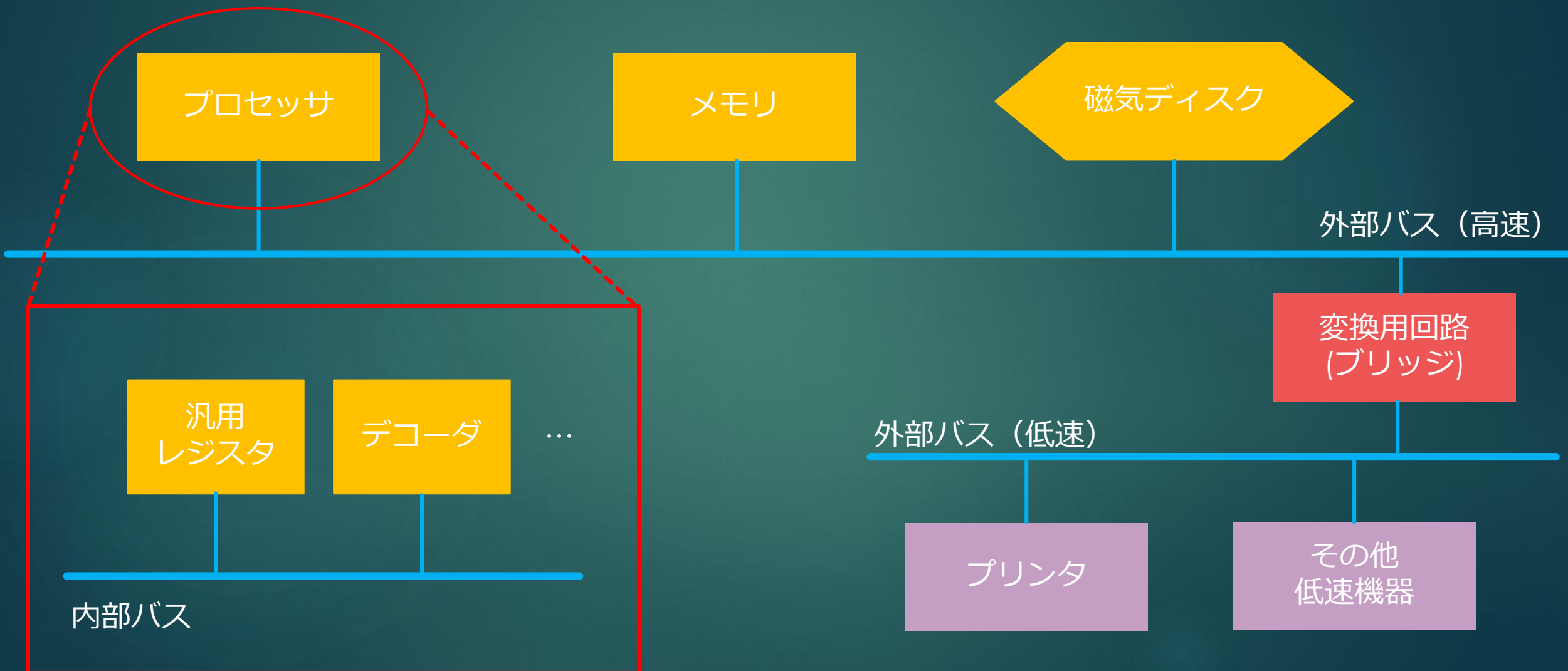
- ▶ 先の構図において一度のアクセスで0~3バンクの内容にアクセスすることができる。
- ▶ 4~7番地に格納された4バイトのデータを1アクセスで取得できる。メモリーインタブルがなければ4~7番地に対して1回ずつのアクセスをしなければならない

7. バスアーキテクチャ

バスの分類

- ▶ **バス**とはデータや制御信号を転送する共通の伝送路のことである
- ▶ バスはプロセッサの機器を接続する**内部バス**と外部機器を接続する**外部バス**に分類することができる。外部バスは高速機器を接続するバスと、低速な機器を接続するバスがあり、両者は**速度変換用のブリッジで接続**されている。

バスの分類



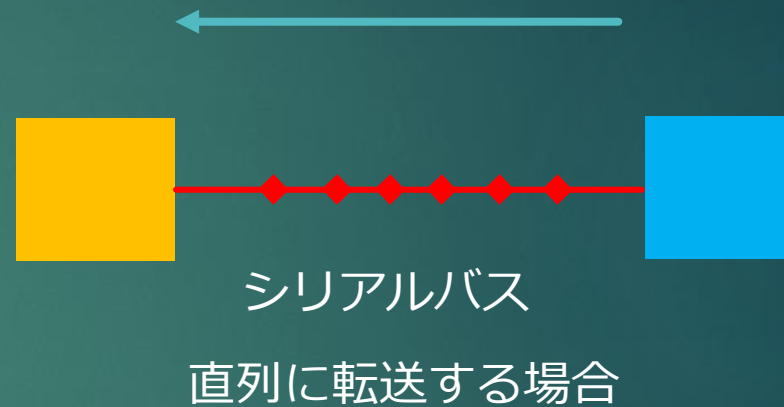
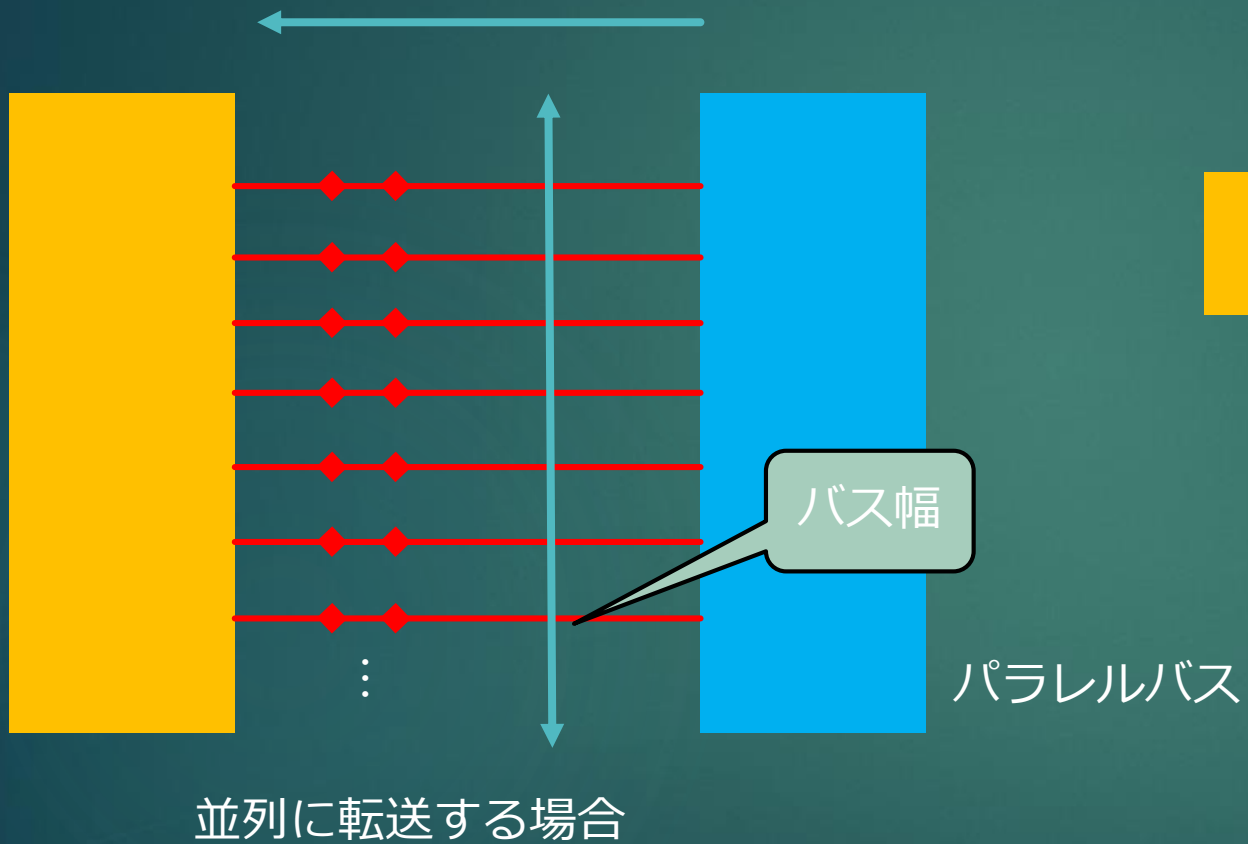
バスの分類

- ▶ 外部バスは、プロセッサやメモリ、グラフィックカードなどを接続する**システムバス**と入出力機器を接続する**入出力バス**に分けることもある

パラレルバス / シリアルバス

- ▶ バスは複数ビットを並列に転送する**パラレルバス**と、1ビットずつ直列に転送する**シリアルバス**に分類することができる。
- ▶ **シリアルバスは高クロックで動作できる**ため、1ビットずつであっても高速に転送することができる。また、構造的に複雑になるパラレルバスに比べ、シリアルバスは単純でコスト的にも有利である。そのため、現在では、「パラレル→シリアル」の流れが進んでいる。

パラレルバス / シリアルバス



バスの種類

- ▶ 代表的なシステムバスの規格は次のものである。

名称	転送方式	最大転送速度	用途
PCI	パラレル	133 / 533M (バイト/秒)	パソコン標準
AGP	パラレル	133M ~ 2.13G (バイト/秒)	グラフィックカード
VME	パラレル	57M (バイト/秒)	ワークステーション、 計測機器
PCIe	シリアル	2 / 4G (ビット/秒)	PCI, AGPの後継

バスの種類

- ▶ **Pc**leはAGP、PCIを後継するシリアルバスで、**PCI Express** と呼ばれる。
- ▶ PCI Express はグラフィックカードの接続に大きなシェアを持つ新しいシリアルバスである。
- ▶ 表の転送速度は1レーン当たりのものであり、複数レーンを束ねることもできる。
- ▶ 例えば16レーンを束ねた方式は PCI Express x 16と表記する。

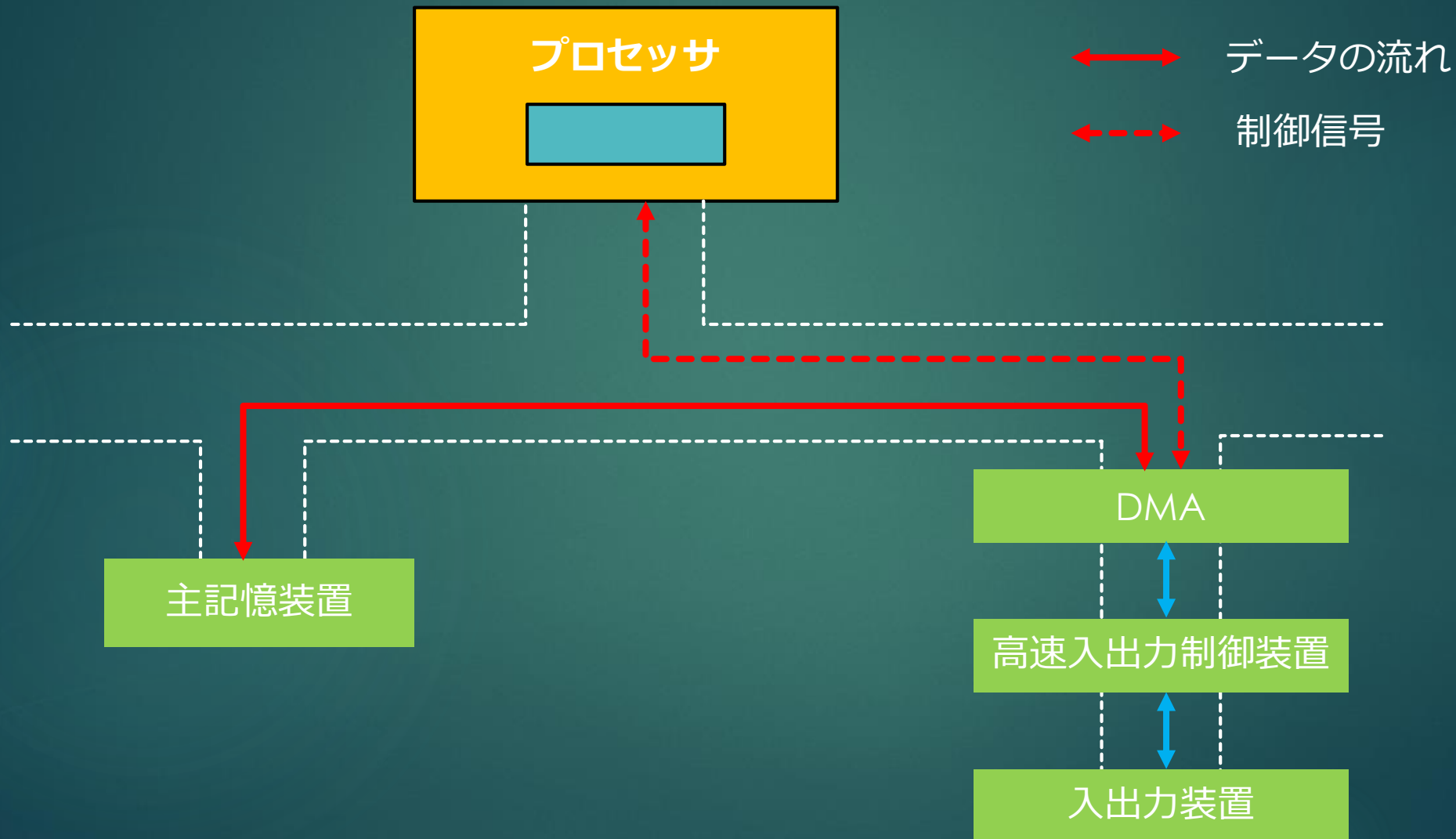
8.入出力アーキテクチャ

入出力制御方式

- ▶ 入出力制御とは、プロセッサと主記憶および入出力機器との間のデータ転送を制御することである。

プログラム制御方式	主記憶装置と入出力装置の間のデータ転送をプロセッサが直接制御する。データ転送はプロセッサを介して行う。
DMA制御方式	DMAコントローラがプロセッサとは独立して 主記憶装置と入出力装置の間のデータ転送を制御 する。
チャネル制御方式	チャネルと呼ばれる入出力専用の処理装置を設ける方式。データ転送は専用のバス経由で行う

入出力制御方式(DMA制御方式)



入出力インターフェース

名称	転送方式	転送速度	特徴
RS-232C	シリアル	115.2kビット/秒	・歴史的な規格
ATA-4	パラレル	33.3Mバイト/秒	・汎用的なパラレルインターフェース ・IDE/EIDEの拡張規格
シリアル ATA	シリアル	1.5Gビット/秒	・ATAをシリアル化し、さらに高速化
SCSI	パラレル	5Mバイト/秒	・ハードディスクなどの接続に用いる ・最大8台までデイジーチェーンで接続可
USB	シリアル	5Gバイト/秒	・汎用的なシリアルインターフェース
IEEE1394	シリアル	100M~3.2Gビット/秒	・デジタルビデオなどのAV機器の接続に用いる
HDMI	シリアル (3レーン)	10.2Gビット/秒	・もともとはAV機器の映像・音声インターフェース ・ディスプレイの接続などに用いる
IrDA	シリアル	115.2k~4Mビット/秒	・赤外線を用いた無線通信規格 ・遮断物があると通信に支障をきたす
Bluetooth	シリアル	1~24Mビット/秒	・2.4Ghz帯域の電波を用いた無線通信規格 ・遮断物があっても問題なく通信できる

入出インターフェース

- ▶ **USB1.0/1.1**は転送速度が12Mビット/秒である
- ▶ **USB2.0**はUSB1.1にハイスピードモード(480Mビット/秒)を追加
- ▶ **USB3.0**はスーパースピードモード(5Gビット/秒)が追加、計4つの転送スピードをもつ。2.0からピン数などが変更されたが物理的な後方互換性をもつ
- ▶ **HDMI**はAV機器用のデジタル映像・音声入出インターフェース規格。映像と音声を1本のケーブルで伝送できる。ディスプレイの接続用途としても用いる。

補助記憶装置

- ▶ 代表的な補助記憶に磁気ディスクとSSD(Solid State Drive)がある。

磁気ディスク	磁気を塗布したディスクに磁気ヘッドでデータを読み書きする補助記憶装置。テラバイト単位の容量をもつものもある。
SSD	記憶媒体にフラッシュメモリを用いた補助記憶装置。数百ギガ単位の容量をもつものが多い。

- ▶ 磁気ディスクはモータで回転させたディスク上を、磁気ヘッドが移動してデータをアクセスする。SSDのアクセスにはそのような機械的な動作が不要なので、高速にアクセスできる。ただし、SSDには書き込み回数の上限があり、磁気ディスクに比べてビット単価が高いなど不利な点もある。

補助記憶装置

- ▶ **ディスクキャッシュ**とは磁気ディスクの高速化技法の一つで、磁気ディスク装置に内蔵されたICメモリを、磁気ディスク装置へのキャッシュとして用い、平均的なアクセス時間を減少させる。

光ディスク

名称	容量	説明
CD-ROM	数百Mバイト	・ 読み出し専用のCD ・ インストールディスクなどの用途が多い
CD-R	数百Mバイト	・ 追記可能なCD
CD-RW	数百Mバイト	・ 消去/記録可能なCD
MO	～数Gバイト	・ データの記憶に磁気を併用する光磁気ディスク ・ データのバックアップ用途に用いられる
DVD-ROM	数Gバイト	・ 読み出し専用のDVD
DVD-R	数Gバイト	・ 追記可能なDVD
DVD-RAM	数Gバイト	・ 消去/記録可能なDVD
Blu-ray Disc	数十Gバイト	・ DVDの大容量化

Blur-ray Disc(BD)は、DVDよりも**波長の短いレーザ光を用いる**ことで、高密度化/大容量化を実現した光ディスク媒体である。ただ、コンピュータの用途として広く普及するに至っていない。

入出力装置

- ▶ 出力装置の代表がディスプレイやプリンタである。特にディスプレイは技術的な発展が目覚ましく、様々な方式が混在している。入力装置はキーボードなどが定番であるが、スマートフォンやパッド型のコンピュータの登場で、ディスプレイから直接入力するタッチパネルが一般的になった。
- ▶ ディスプレイの方式は次のようなものがある。

CRTディスプレイ	ブラウン管を用いたディスプレイ
液晶ディスプレイ	「電圧をかけると分子の向きが変化する」という液晶の性質を利用して表示を行う。光の透過を画素ごとに制御し、カラーフィルタを用いて色を表現する。
有機ELディスプレイ	電圧をかけると自ら発行する有機化合物を画素に用いたディスプレイ。バックライトなどの光源が不要、応答速度に優れている、視野角が広い、消費電力が少ない、といった特徴がある。

カラーの表現方法
・3色の発光素子をもつ→有機EL
・カラーフィルタ→液晶
※カラーフィルタを採用する有機ELディスプレイもある

入出力装置（プリンタの方式）

名称	特徴	
ドットインパクトプリンタ	インクリボンをヘッドによって打ちつける。複写紙を用いた伝票複写が可能	
サーマルプリンタ	感熱	熱によって変色する専用紙を、発熱ヘッドで感熱させる
	熱転写	熱で溶融するインクリボンを用い、発熱ヘッドで紙に転写する
インクジェットプリンタ	ノズルから細かいインクを噴出する	
レーザ(LED)プリンタ	コピー機と同様の原理で、トナーを紙に転写する。印字品質・スピードに優れるが、消費電力が大きい。	

入出力装置（タッチパネルの方式）

- ▶ タッチパネルは指やペンで触れることで情報を入力するパネルである。タッチの検出方式により次のように分類することができる。

静電気式	タッチ時の微弱電流を検知する方式。 複数個所のタッチを検出できるため、 ピンチイン/アウトなどの マルチタッチ入力 が可能
感圧式	タッチ時の圧力を検知する方式

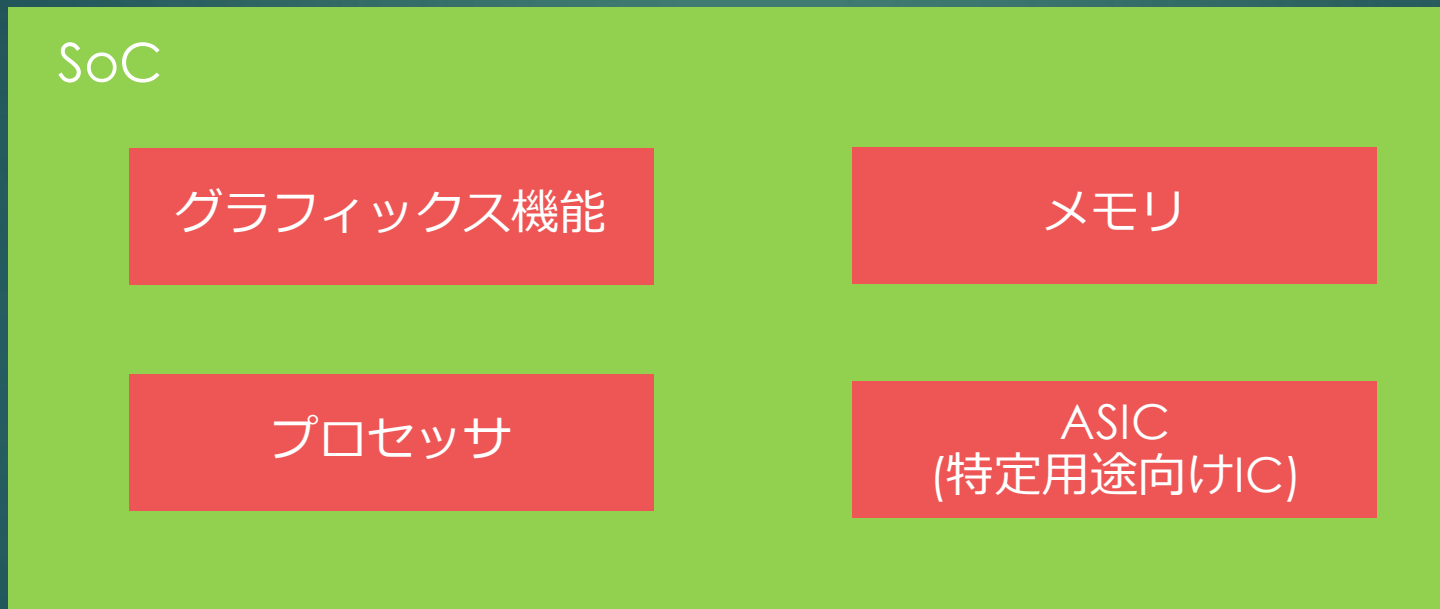
9. 組み込みシステムのハードウェア要素

LSI (Large Scale Integration)

- ▶ ハードウェアは、目的に応じて複雑な機能を果たさなければならない。その実現のため、多くの半導体素子を集積した集積回路(IC)が用いられている。ICの中でも集積度の高いものが**LSI**であり、組み込みシステムの基本的な構成要素となっている。
- ▶ LSIを含むIC製品には**高電圧により素子が破壊される(ESD)**ことがある。そのため製造現場においてはさまざまな静電気対策を施している。
- ▶ **ESD(Electrostatic Discharge)**とは静電気により半導体素子が破壊される現象のことである。

SoC(System on a Chip)

- ▶ **SoC**は、従来はボード所で実装していた**一連の機能を1チップに集約**したLSIのことである。集約による小型化、高速化、低消費電力化に加え、**量産効果による低価格化**など様々な利点を得ることができる。



LSI (Large Scale Integration)

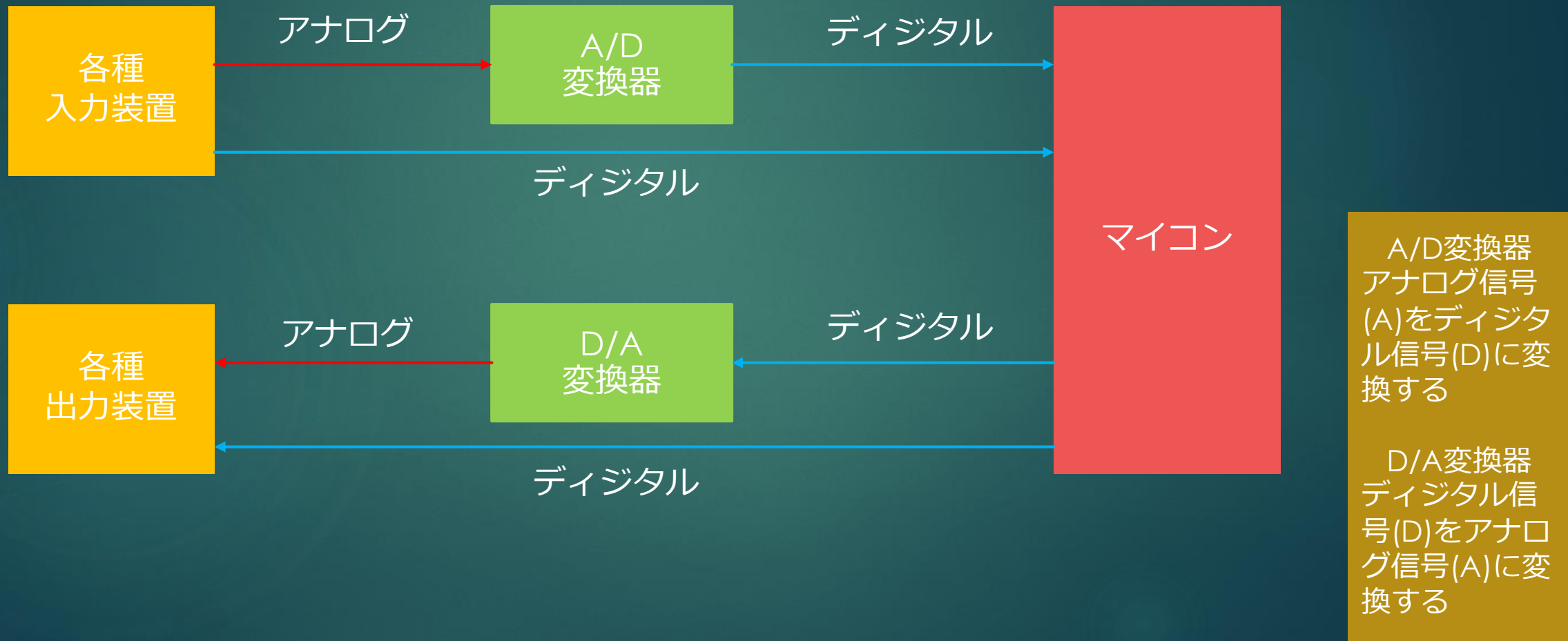
- ▶ **SiP (System in Package)** は複数のLSIチップを一つのパッケージにまとめたものである。
- ▶ **システムLSI** は **複数の機能を1チップに集約** した多機能LSIをシステムLSIという。組込システムで用いるシステムLSIには、主要回路がすべて1チップに集約している。SoCとほぼ同じ意味で用いられることも多いが広義のシステムLSIにはSiPを含めることもある。

LSI (Large Scale Integration)

- ▶ **DSP (Digital Signal Processor)**とは**デジタル信号をリアルタイムで処理**する専用のプロセッサである。動画や音声の再生では、圧縮ファイルをリアルタイムに展開し再生する必要がある。そのため、デジタル信号を高速に処理する専用のDSPを用いる。
- ▶ **FPGA (Field Programmable Gate Array)**は回路を**プログラムにより再構成できるIC**のことである。製造時に内容が固定される**ASIC (Application Specific IC)**に比べ、柔軟性に優れ、製品のアップデートにも対応しやすいことが特徴である。

入出力

- ▶ 組込システムでは以下のような構成をとる



入出力

- ▶ 入力には各種センサ類やスイッチで行う。センサ類には、**フォトダイオード**や**フォトランジスタ**などの光センサ、**CCD**や**CMOSセンサ**などのイメージセンサ、エアコンの温度調節に用いる温度センサなどがある。
- ▶ センサの中にはアナログ信号が入力されるものもある。アナログ信号は**A/D変換器(ADC)**でデジタル信号に変換し、マイコンに入力する。
- ▶ 出力はLEDやディスプレイ、湯沸かしポットではヒータ、産業用のロボットではアームを動かすための**アクチュエータ**などがある。音声などのアナログ信号を出力する場合は、デジタル信号を**D/A変換器**を通してアナログ信号に変換し、スピーカなどから出力する。

チャタリング
スイッチなどで、
機械的振動により
ON/OFFが短時間
のうちに繰り返
される現象

アクチュエータ
モータなど、機
械や機構を物理
的に動かすため
の駆動装置

A/D, D/A変換器と最大電圧

- ▶ A/D, D/A変換器は種類によって**ビット数（量子化ビット数）と最大電圧が定まっている**。たとえば最大電圧が2.5Vでビット数が8ビットであれば、2.5Vの電圧を256刻みで分解できることを表す。このとき、**分解能（ビット1あたりのアナログ値）**はおよそ10ミリVとなる。
- ▶ 分解能の計算式は以下ようになる。

$$\text{分解能} = \frac{\text{最大電圧}}{2^{\text{ビット数}}}$$

A/D, D/A変換器と最大電圧

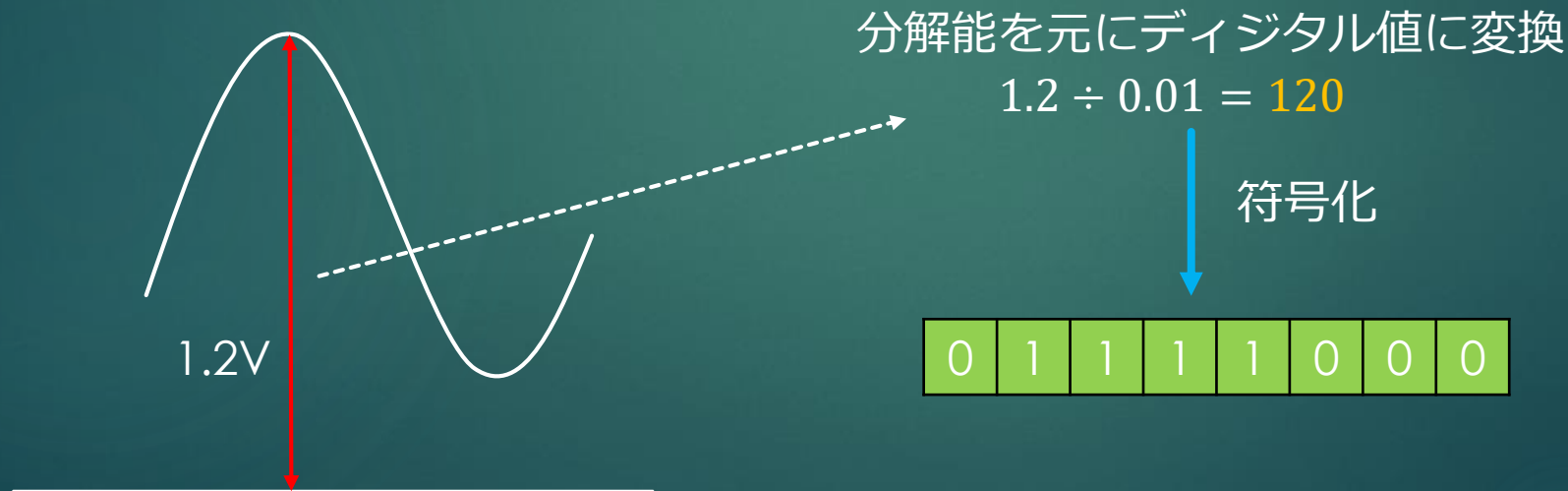
- ▶ A/D変換器の出力値は、計測値（アナログ値）と分解能、ビット数で定まる。たとえば分解能10ミリVの8ビットA/D変換器で1.2Vの電圧を計測したとき

$$1.2 \div 0.01 = 120 = (78)_{16}$$

を出力する。逆に同じ仕様のD/A変換器を用いれば、ディジタル値120は、

$$120 \times 0.01 = 1.2\text{V}$$

の電圧を出力する。



- A/D変換の出力値
= アナログ値 ÷ 分解能
- D/A変換の出力値
= ディジタル値 × 分解能

PCM(Pulse Code Modulation)

- ▶ **PCM**は**音声をデジタルデータに変換**する方式の一つである。一定間隔で音声を**サンプリング（標本化）**し、A/D変換でデジタル値に変換する。1秒間に行うサンプリングの回数を**サンプリング周波数**と呼ぶ。例えば、量子化ビット数が24、サンプリング周波数が48kHzステレオ(2系統)という条件でサンプリングしたとき、1秒間のデジタルデータ量は $24 \times 48000 \times 2(\text{ビット}) = 2304(\text{kビット}) = 288(\text{kバイト})$ となる。

標本化
(サンプリング)
音の信号を一定の間
隔で切り出すこと

1秒間の音声のデータ量は

量子化ビット数×サンプリング周波数[ビット]

で求まる

標本化定理

- ▶ **原音の周波数の2倍のサンプリング周波数**を用いてデジタル化すれば、デジタルデータから原音を**完全に復元**できる。これを**標本化定理**と呼ぶ。
- ▶ 例えば、最高20kHzである原音を完全に復元するためには40kHzのサンプリング周波数を用いてデジタル化すればよいことになる。

機械制御方式

- ▶ 産業用ロボットなどに用いられる組込システムは、製造機械の動作を制御する。制御方式には次のようなものがある。

フィードバック制御	外乱による機械の状態を検出・評価し、目標値との差を制御量に反映させることで、より高精度な制御を行う方式
フィードフォワード制御	外乱を直接検知し、その影響を少なくするよう制御量を定める制御方式
シーケンス制御	定められた手順通りに制御を行う方式

- ▶ シーケンス制御に用いられる制御装置を**PLC(Programmable Logic Controller)**、PLCが実行する制御手順を**シーケンスプログラム**と呼ぶ。シーケンスプログラムは主に**ラダー図**と呼ばれる記述法で表される。

ラダー図は、論理回路を記述する手法の一つで、図が梯子（ラダー）のように見えることからこの名称がついた

ワンチップマイコン

- ▶ 入出力インターフェースやプログラムメモリなどを一つのICチップの中に詰め込んだものを**ワンチップマイコン**と呼ぶ。
- ▶ クロックは、ワンチップマイコンに内蔵された**クロックジェネレータ**が供給する。この時、高速に動作するCPUには高い周波数を、低速な機器には低い周波数を供給するので、基準となるクロック周波数をPLLや分周器で調節する必要がある。
- ▶ **PLL**は**入力のN倍の周波数**を、**分周器**は**1/N倍の周波数**のクロックを出力する。

リーク電流

- ▶ 半導体の微細化が進むに伴い、本来電流が流れない回路に絶縁体越しに電流が流れる現象が生じた。これを**リーク電流**と呼ぶ。
- ▶ リーク電流が流れると、**回路の誤作動や消費電力の増加、発熱につながる**ため対処が必要である

クロックゲーティング

- ▶ 組込システムの中には、間欠的に動作すればよいというものもある。たとえばデジタル時計あれば、1秒間に1回動作すれば十分である。そのような場合は、動作するときのみクロックを供給し**動作しない間はクロック供給を停止する**ことで、消費電力を節約する。そのような省電力技術を**クロックゲーティング**と呼ぶ。ボタン電池で長期間動作するような機器には、たいていこの機構が組み込まれている。

パワーゲーティング

- ▶ 組込システム内のすべてのブロックが常に動作しているわけではない。**パワーゲーティング**は、**動作する必要がない回路ブロックへの電源供給を遮断することによって、消費電力を減らす**。パワーゲーティングは**リーク電流対処**として有効である。

タイマ / カウンタ

WDT(ウォッチドッグタイマ)

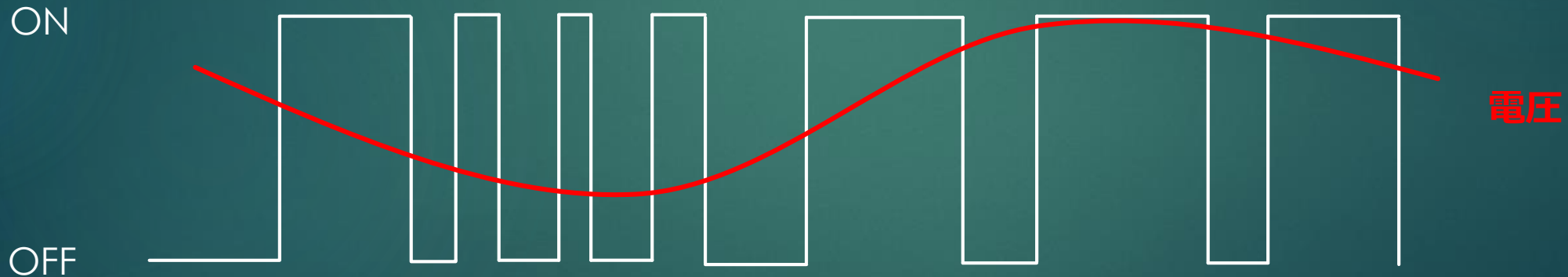
- ▶ 内部カウンタをクロックでカウントアップすることで、時間を計測できる。そのような機構を**タイマ / カウンタ**と総称している。
- ▶ **WDT**は**プログラムの暴走を防ぐ手段**で、タイマの一種である。WDTはあらかじめ時間間隔を設定しておく、プログラムが定期的にタイマをクリアする。**プログラムが暴走すると、定期的なタイマのクリア処理が実行されず、結果としてWDTに設定した時間を超えてしまうことになる。**このとき、システムに異常が発生したとみなして、システムに通知する。

フラッシュメモリの採用

- ▶ ワンチップマイコンの内蔵メモリとして**フラッシュメモリ**が採用されることが多くなっている。フラッシュメモリは電氣的に内容を書き換えることができるため、その採用により出荷後であってもソフトウェアの書換えが可能になった。

PWM(Pulse Width Modulation)

- ▶ **PWM**は半導体を使った電力制御方法の一つで、オンとオフの間隔を変化させることで、出力電力を制御する。**電圧が高いほどオンの間隔が長くなる。**



ジャイロセンサ

- ▶ **ジャイロセンサ**は物体の角度や角速度、各加速度を検出するセンサで、スマートフォンなどに搭載されている。スマートフォンがどれだけ傾いているか、どれだけの早さや勢いで傾きを変えているか、振動しているかどうかなどを検出できる。