

1) コンピュータを高速化したい。まずは、コンピュータの基本構造は変えないで、クロックの速さを速くすることで、1つ1つの命令を処理する速さを速くする方法が考えられる。この方法の問題点は何だろうか？

2) コンピュータを高速化するために、今まで順番に1つずつやっていたことを、複数を同時に行うようにする「並列化」が考えられる。命令パイプラインはその1つの方法といえる。

ア) 命令パイプラインの構造を絵に描いて、その上で命令がどのように実行されるのか、時間を追って(t, t+1, t+2, ...)説明せよ。

イ) 命令パイプラインのステージ数(段数)がSである時、N個の命令が通過するのに要する時間を、図を描いて示せ

ウ) 命令の個数Nが、無限に大きくなれば、パイプラインの流れ始めの時点や、最後の命令の通過時点での「あき」が無くなるはずである。この状態での「加速率」はいくらか？

$$\text{加速率} = (\text{パイプライン無しでかかる時間}) / (\text{パイプラインありでかかる時間}) \quad \text{とする。}$$

エ) パイプラインのストール(乱れ)は、加速率を低下させる。次の2つの例において、どうして乱れが起こるかを、図を描いて説明せよ。但し、パイプラインの構造は下記の通りとする。

演算命令	命令 読出	命令 解釈	命令実行		
			実効アド レス計算	メモリ アクセス	計算結果 書込み

分岐命令	命令 読出	命令 解釈	命令実行
			アドレス計 算・ジャンプ

**データハザードの例**  
GR3、GR4、メモリ 234 番地には予め値が入っている

```

    ADDA GR3, 234    GR3 ← GR3 + (234)
    LD   GR4, X, 456 GR4 ← (456 + (GR4))
    
```

**制御ハザードの例**  
GR3、メモリ 234 番地には予め値が入っている

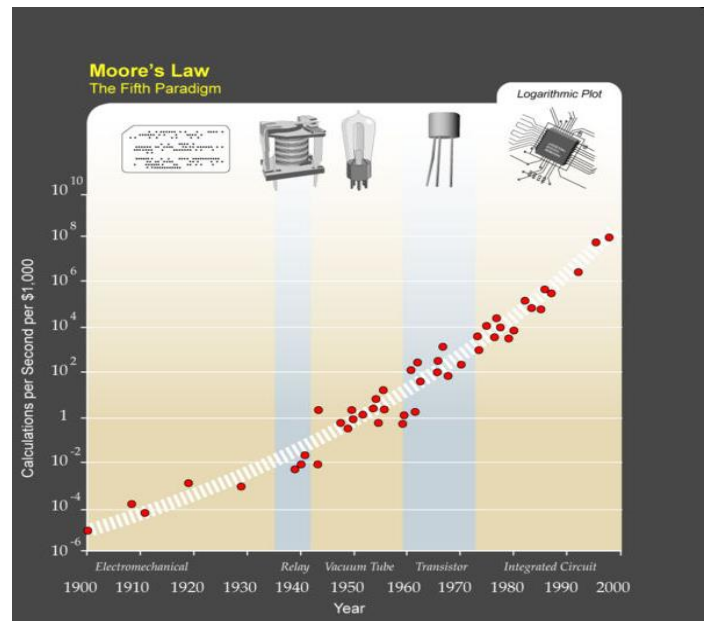
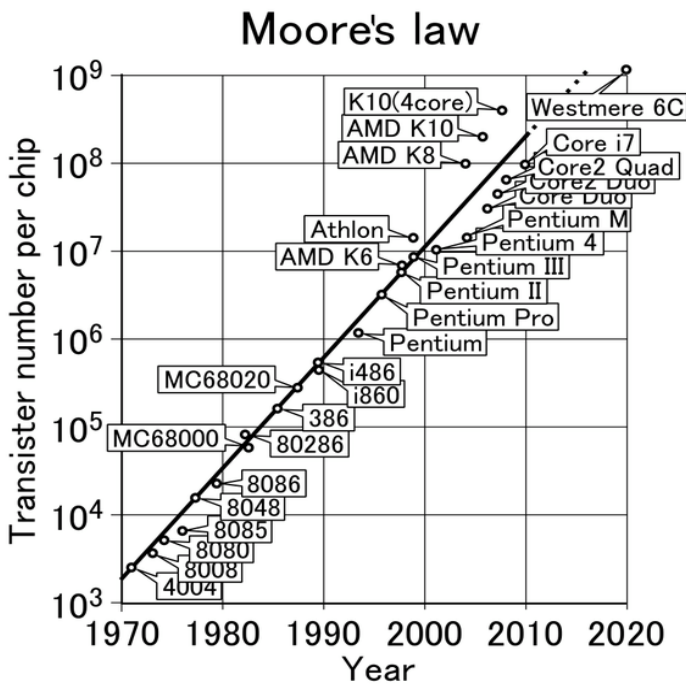
```

    CPA GR3, 234    GR3 と (234) 比較 (結果が正)
    JPL L1          比較結果が正なら jump
                   負の時の処理
    ○○○○○
    ○○○○○
    JUMP L2
L1  ×××××        正の時の処理
    ×××××
L2  △△△△△        ここで合流
    
```

3) (予習資料にない話) ムーア(Moore)の法則

「LSIの密度は18か月ごとに2倍になる」 (Gordon Moore, 1965年) (経験則)

当時は「予測」 ⇒ 実際に2000年過ぎまでほぼ成り立った



[http://ja.wikipedia.org/wiki/ファイル:Moore's\\_law\\_\(1970-2010\).PNG](http://ja.wikipedia.org/wiki/ファイル:Moore's_law_(1970-2010).PNG)

<http://ja.wikipedia.org/wiki/ファイル:PPTMoore'sLawai.jpg>

また、右の図にあるように、処理能力(速度)も1960~1990年で約10<sup>6</sup>倍、つまり1.5年で2倍程度の伸びである。

しかし、最近 LSI 集積密度の伸びが限界に近いといわれている。現在、加工精度はたとえば Intel Skylake(2015/8)では14ナノメートルで、光加工(光波長400~700ナノメートル)をはるかに下回り、電子線加工を行っているが、そもそもSiの格子定数(単位格子の1辺の長さ)が0.35ナノメートルなので、30格子ぐらいを加工していることになり、限度が見えている。

なお、ムーアの法則は素子技術をまたがって乗り越えることで成り立ってきた経緯があるので、今後まったく別の原理による計算機ができて、法則自身は成り立ち続けるかもしれない。

4) (予習にはない話) 並列処理 ~ 現在の「苦闘」

発想: CPUで処理(計算)をしているのは、演算器1か所だけ。もっと同時に(並列に)計算できないものか?

(発想の裏) 化学反応は、たとえば液中で多数の分子が同時並列に反応していて平衡状態を作る。生物も同じ。

いろいろな並列の可能性:

ア) パイプライン

イ) 「マルチコア」 ~ CPUをたくさん並べて、それぞれにプログラムを走らせる。パソコンやスマホでもマルチコア  
これの巨大なものが「京」のようなスーパーコンピュータ

ウ) 「裏」で言ったような、並列の現象の利用ができないか? 化学コンピュータ? 生物コンピュータ?

最近話題になっているのが「量子コンピュータ」

「アムダールの法則」 処理が(直列部+並列部)から成るとき、仮に並列部が並列処理されて殆ど時間=0になると、直列部が全体の処理時間を決める。たとえば、直列部が10%のとき、並列度を十分上げて並列部が殆ど0で計算できても、全体の処理時間は元の時間の1/10にしかならない。