

1) コンピュータの3つの要素の、名称と機能を説明せよ

①(名称).....(機能).....

.....

②(名称).....(機能).....

.....

③(名称).....(機能).....

.....

.....

2) 命令の例を3つ挙げて、それぞれの機能(何をどうする命令か)を説明せよ (名称は不正確でよい)

① (.....)命令 機能:.....

.....

.....

② (.....)命令 機能:.....

.....

.....

③ (.....)命令 機能:.....

.....

.....

.....

3) ノイマン型コンピュータの3つの要件を挙げ、その内容(どういうことか)と、メリット(そうでない時に比べて何がよくなるか)を説明せよ。名称は固定していないのでアバウトでよい

① (要件の名称).....

(内容).....

.....

(メリット).....

.....

② (要件の名称).....

(内容).....

.....

(メリット).....

.....

③ (要件の名称).....

(内容)

(メリット)

4) 1つの命令が実行される時の、内部の処理手順(4つのステップ)を挙げ、それぞれが何をするステップであるか(動作記述)を説明せよ。(ステップの名称をそのまま動作記述に写したものは不可で、名称よりはさらに具体的な記述がなければならないものとする)

(ステップ1)

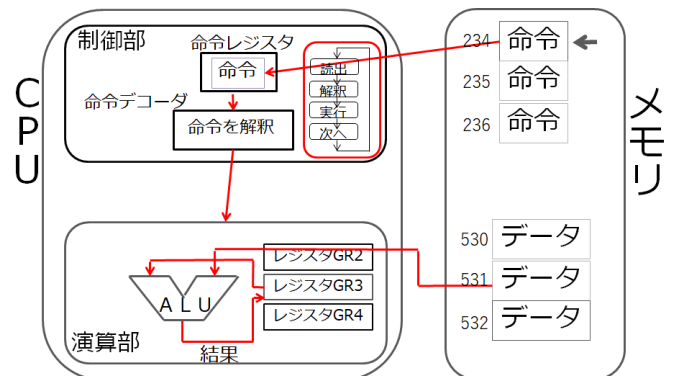
(ステップ2)

(ステップ3)

(ステップ4) $PC \leftarrow PC + 1$ プログラムカウンタPCを1つ進める。次の命令実行サイクルで、1つ増えたPCの値をアドレスとして命令を読み出すので、次の命令に進むことになる。ノイマン型の「逐次実行」を実現している。

5) (時間があれば) 命令実行の実際のイメージ

(あ)メインメモリ中の 234 番地に、命令 `ADDA GR3 531` が置いてあるとしよう。この命令を実行するときの様子を具体的に書き出してみよう。



① ステップ 1()

読み出す命令の場所の指定は()にあり、この例では()が入っている。読み出した結果、()番地の内容は()だった。

② ステップ 2()

読みだした内容は単なる()数だが、それを「命令やそれに関する情報が()化されて書かれている」ものとして、「解釈」する。解釈に用いる符号化のルール(2進数がどんな意味を持つか)は(たぶん)次回に説明する。

ここではその2進数が、命令 `ADDA GR3 531` と解釈できたとする。この命令は、

<レジスタ GR3 の内容と、メインメモリ 531 番地の内容を読み出して、加算して、結果をレジスタ GR3 に書きこむ> というもの(という指示をするもの)である。

③ ステップ 3()

解釈の結果に従って、()部は他の部分に指示を出してゆく。この命令 `ADDA GR3 531` は次の 2 ステップの指示を出すことになるだろう。(1)メインメモリの()番地を読み出す。(2)レジスタ GR3 の内容と(1)の結果を加算器に与え、その出力をレジスタ GR3 に書きこむ。

④ ステップ 4()

①で使った、「読み出す命令の場所」を指定するレジスタ()を1つ進める。

①に戻る。

(い) 右上の図と「ノイマン型」の 3 つの要件との関係を説明してみよう。要件①() 要件②() 要件③()