

[1] ア)イ)は2の補数を作る手続き 第2回予習ビデオ「マイナスの数の表現」を参照

ウ)エ)は、左端ビットが1なので負の数(問題に「符号付き」) 10進への変換は第2回ビデオ「マイナスの数おまけ」を参照
オ)カ)キ)は、第3回ビデオ「小数の表現-固定小数点」を参照。なお、 $0.2_{10} = 0.1_{10} \times 2 = 0.1_{10}$ を右へ1ビットシフト
ク)は、見たことがない問題だと思う。浮動小数点の仕組みをわかっているか見たかった。

仮数部が0. $101_2 = 0.625_{10}$ 、指数部が $10_2 = 2_{10}$ というので、表す数は、 $0.625 \times 2^2 = 0.625 \times 4 = 2.5$

[2] ① 符号付き32ビットの最大数は、 $0111\ 1111\ 1111\ 1111\ 1111\ 1111\ 1111\ 1111_2$ である。この数に1を足すと、左端のビットが1になってしまうので、負数になる。で、これは $2^{31}-1$ に当たる。これを10進数に直せと言っているのだが、丁寧に 2^{31} を計算せずに、 $2^{10} \div 10^3$ の関係を使ってよいと言っているので、 $2^{31} = 2 \times (2^{10})^3 \div 2 \times (10^3)^3 = 2 \times 10^9$ となる。なお、 $2^{31}-1$ の(-1)の部分は 2×10^9 に比べて非常に小さいので無視してよい。

② 最小の数は、 $1000\ 0000\ 0000\ 0000\ 0000\ 0000\ 0000\ 0001_2$ である。何人かが $2^{-31}-1$ のように肩の31のところマイナスをつけていたが、これは「絶対値としては小さい」=「正でゼロに近い」のだが、符号付32ビット整数は「(絶対値の大きい)マイナスの数」を表すことができ、そちらの方が「小さい」。単純には、①で考えた最大の正の数の符号をマイナスにしたものである。上記の2進表現 $1000\ 0000\ 0000\ 0000\ 0000\ 0000\ 0000\ 0001_2$ を符号反転すると①の数になる。

[3] ア) 1桁の2進加算回路(フルアダー)の、入力と出力が何であって、どういう真理値表で表現されるか、という問題である。第4回の予習ビデオ「2進加算の原理」を参照。

イ) 真理値表で表された論理を、AND/OR/NOT の組合せで作るという問題で、第1回「任意の論理演算とその合成」を参照

ウ)エ) 複数桁の加算回路を、1桁の加算回路を組み合わせて作る。第4回ビデオ「2進加減算の回路」を参照

[4] 1つの命令の実行ステップを説明する問題。第5回ビデオ「命令が実行される仕組み」、「ノイマン型コンピュータ」参照

① どこから (A) メインメモリ(主記憶) の (B) CPU のプログラムカウンタ PC で指定されたアドレスの場所 から
どこへ (A) CPU (中央処理装置) の (B) 「制御部」 へ

② 命令に含まれている、命令(OP)部、オペランド部などを識別し取り出す。

回答として「(読み出した)命令を解釈すること」は、認めなかった。理由は、問題文をそのまま写しただけだから。

③ 命令(OP)部に指定された動作をさせる信号を、演算部やメモリに出して、動作させる。

回答として「(解釈した)命令を実行すること」は、認めなかった。理由は、問題文をそのまま写しただけだから。

④ PC(プログラムカウンタ)は、現在実行中の(または次に実行する)命令の、メモリ上のアドレスを保持するものである。

(注意) PC 自体はアドレス(の数値)を保持するだけで、実行をするわけではない。

[5] 第6回予習ビデオ「アドレッシング」を参照。(ア) 1003 (イ) 1007 (ウ) 1007 (エ) 1000 (オ) 4

[6] 第6回予習ビデオ「命令の実行性能」参照。ア) $2 \times 0.6 + 3 \times 0.2 + 5 \times 0.2 = 2.8$

イ) 「頻度が違うので掛ければよい」という説明は不十分で、なぜ「掛ければよい」のかを説明してほしい。たとえば、プログラム全体でかかる時間は、それぞれの命令にかかるクロック数の総和である。表に示された頻度で命令が出現するのであれば、各種別の命令にかかるクロック数×その命令の実行回数になるから、これをプログラム全体の命令数で割ると、各種別の命令にかかるクロック数×その命令の出現頻度になる。だから出現頻度による加重平均を用いる。

ウ) 1クロックの時間 = $1 / (3 \times 10^9)$ (秒) = $(1/3) \times 10^{-9}$ 。 1命令当たりの平均実行時間 = CPI × (1クロックの時間) = 0.933 (ナノ秒)

エ) MIPS値 = $(1 / (1命令当たりの実行時間)) / 10^6$ (MIPSは 10^6 が単位) = 1.072×10^3 (MIPS)

[7] (ア) 命令の意味の書き方は、第7回予習ビデオ「COMET-IIのプログラミング」26ページ目ぐらいを参照。

(イ) AもBも、計算結果は同じで、 $S=X+Y+Z+W$

(ウ) 大きな違いは、プログラム実行の時に実行する命令の数が、Aは5つに対して、Bは8つで、Bの方が多い。もしすべての命令のCPI(1つ1つの命令を実行するクロック数)が同じであれば、Bの方が余分に時間がかかる(遅い)。

(注意) なお、「すべての命令のCPIが同じである」とは、問題文中には書いていない。命令によって実行クロック数が異なるケースはあり得るので、回答するときに「もし同じであると仮定すれば」という断りを入れるのがよいだろう。今後大学で勉強する中で、問いが完全でない問題が大いに出てくると思うので、その時はきちんと自分で設定すること。

[8] RISCとCISC 第6回の予習ビデオ「CISCとRISC」を参照のこと。

(ア) 命令体系の作り方(考え方)の違いで、

RISCは、簡単な命令、簡単な体系を使って、複雑なことは命令の組み合わせで実現する

CISCは、複雑な命令、複雑な体系を使って、複雑なことを1つの命令で実現する(複雑なことをできる命令を多数用意する)という感じ。(予習ビデオ5ページ)

(イ) 比較の表は、予習ビデオ27ページ参照

(ウ) たとえば、

同じことをするのに、CISCだと1つ1つの命令のすることが多いので、全体として少ない命令ステップ数で実現できる。RISCだと1つ1つの命令が単純な今年ができないので、全体として命令ステップ数は多くなる。しかし、CISCは1つの命令にかかる時間が長くなる(複雑だから)のに対して、RISCは1つの命令の実行が速い(簡単だから)。トータルの時間(=命令ステップ数×1つの命令にかかる時間)は、どちらがよいともいえない。

もっとほかの比較もできるだろう。

[9] 第7回の予習ビデオ「条件分岐とIF文」の例題を参照。

