

## 2019 年度試験問題と解答例

1-1. 8 番地の内容と A、C 番地の内容を B とする時、 $(A+B) \text{ XOR } (A-B)$  を計算する RV32I アセンブラプログラムを書きなさい。

```
lw x3,x0,8
lw x4,x0,0xC
add x5,x3,x4
sub x6,x3,x4
xor x5,x6,x5 答えはここでは x5 に入れている
```

1-2. A が 0x90000008 番地、B が 0x9000000C 番地であった。1-1 と同じ RV32I プログラムを書きなさい。

```
lui x7,x0,0x90000
lw x3,x7,8
lw x4,x7,0xC
add x5,x3,x4
sub x6,x3,x4
xor x5,x6,x5
```

2. ボタン入力  $b$ 、 $r$  は、押すと 1、放すと 0 になる。 $r$  を押すと  $p$  は 0 になり、 $b$  を 2 回目に押した際に出力  $p$  が 1 になる module two(input b,r,clk, output p); を Verilog HDL で書け。なお clk はボタンに比べて十分の高速なクロックが入り、チャタリング (スイッチを押した時のばたつき) は、考慮しなくて良いとする。(ヒント: 状態遷移図に基づくのが王道。clk を無視して  $b$  をクロックとして扱うのは邪道だが動けば正解とする。王道で動くものにはボーナスを差し上げる)

```
module two( input b,r, clk, output p);
  reg[3:0] stat;
  always @(posedge clk or posedge r) begin
    if( r ) stat <= 4'b0001;
    else case (stat)
      4'b0001: if(b) stat <= 4'b0010;
      4'b0010: if(!b) stat <= 4'b0100;
      4'b0100: if(b) stat <= 4'b1000;
    endcase
  end
  assign p = stat[3];
endmodule
```

3.  $x5$  の内容が 4 の倍数ならば  $x4$  に 1、そうでなければ 0 を設定するサブルーチン four を RV32I アセンブラで書け。

4 の倍数は下 2 桁が 0 なので  $2'b11=3$  と AND を取って 0 かどうか調べれば良い。超簡単なはずなのに、

```
andi x5,x5,3
beq x5,x0,multi4
addi x4,x0,0
jalr x0,x2,0 // jr x2
multi4: addi x4,x0,1
jalr x0,x2,0 // jr x2
```

もちろん、4を引き続けて0になれば4の倍数、マイナスになってしまえばそうでない、というのもアリで正しい答えには満点をあげましたが、時間がかかるよね。

4. 上記サブルーチン `four` を用いて、0番地からの100個の数字の中から4の倍数を数えて `x3` に答えを入れるプログラムをRV32Iアセンブラで書け。(実はサブルーチンを使うまでもないのだが、折角作ったので使ってくださいませ。スタックへのレジスタ保存は必要ないです。)

```
    addi x6,x0,400    // pointer 兼 counter
    add x3,x0,x0      //答
lp:  addi x6,x6,-4    // pointer decrement
     lw x5,x6,0      //データ読み出し
     jal x2,four     // four を読み出してチェック
     beq x4,x0,skip  //4 の倍数でなければスキップ
     addi x3,x3,1    // 4 の倍数ならばカウントアップ
skip: bng x6,x0,lp   // ループ
end: beq x0,x0,end
```

`four` 自体が正解でなくてもメインルーチンが正解ならば正解とした。

5. RISC-Vの以下の疑似命令は、実際の命令を用いてどのように実現されているかを示せ。

① `jr x1` (レジスタ間接ジャンプ) ② `li x1,100` (イミディエイト値をレジスタに入れる)

```
jalr x0,x1,0          addi x1,x0,100
```

6. マルチサイクル版のRV32Iでは、メモリアクセスに4クロック、その他は3クロックを必要とする。メモリアクセス命令の生起確率を25%とすると、CPIはいくつになるか? またクリティカルパスが2nsecになった場合、クリティカルパス6nsecのシングルサイクル版と比較すると、どちらがどれだけ速いか?

マルチサイクル版  $CPI = 0.25 \times 4 + 0.75 \times 3 = 3.25$

$(2 \times 3.25) / 6 = 1.083$  倍シングルサイクル版が速い

7. 1Mbyteの主記憶に対して、8Kbyteのキャッシュを設ける。ブロックサイズを32byteとした場合、2way set associative cache, 4way set associative cacheのディレクトリ構造はどのようになるかを示せ。

8Kbyte=2の13乗、32byte=2の5乗 2の8乗個キャッシュ上にブロックが搭載できる。indexは8ビット。Tagは2の20乗の主記憶に対して20-8-5=7ビット

2-way set associative Index 7 bit (128 エントリ) Tag (幅) 8bit X 2

4-way set associative Index 6bit (64 エントリ) Tag (幅) 9bit X 4

これは、予告もしたんだし、去年の数値を変えたただけなので間違わないで欲しかった!

8. `lui x1, 0x12875` `sw x1,x0,4` `lb x2,x0,6` `lbu x3,x0,6` を順に実行した。`x1,x2,x3`の値はどのようになるかを示せ。

`x1`は12875000になり、これが4番地から書かれることになる。リトルエンディアンだと、6番地は下位から3つ目のByteで87になる。`lb`は符号拡張するので0xfffff87 `lbu`はゼロ拡張なので0x00000087

見え見えの問題だと思ったのだが案外正答が少なかったので、部分点を大量に上げた。

9. RISC-V はミスアラインメントを許している。このことの利点と欠点について簡単に説明せよ。

利点：メモリの利用効率が良い。レガシーコードがコンパイルし直さないで使える

欠点：動作時間が掛かる。ハードウェアが複雑化する。

10. 他の RISC プロセッサ (MIPS や ARM) に比べた RISC-V の特徴を簡単に説明せよ。

基本命令セットからモジュール的に命令を拡張して行くことができる。従来の RISC の反省を生かしてあるため、実装と命令セットが分離しており、高速動作が可能である。

後半は、異論もあります。ちなみにレジスタ-レジスタアーキテクチャであるとか、32 ビット単一長である、とかは、他の RISC と共通しているので、正解としていません。

今回は RISC-V の登場によってやや傾向が変わったために、難しかったかもしれません。部分点を大量に上げました。しかし、1, 6, 7 を間違った方は反省が必要です。少しは試験勉強してください。