情報工学科 計算機構成 2014 年度 期末試験 (担当:天野 持ち込み何でも可)

POCO の命令セット、ハードウェア構成は 2 ページ目以降に示しますが、問題中に不明な点がある場合や手元に資料がない場合は、各自判断して答え、その旨を記して下さい。注意：問題用紙は 2 ページあります。

1. 0 番地に A, 1 番地に B が入っている。$(A-B)OR(A>>1)$ を計算し、答えを 2 番地に入れる POCO のプログラムをアセンブリ言語で記述せよ。
2. 問題 1 のプログラムの最初の 4 行を機械語に変換せよ。
3. A, B それぞれ 2bit の入力のうち A が B より大きい場合に出力 Y を 1 にするモジュールを Verilog HDL で記述せよ。
4. 0 番地から 99 番地までの 100 個の数を、100 番地から 199 番地に移動するプログラムを POCO のアセンブラで書け。
5. サブルーチン div は、r0/r1 を実行し、答えを r2、余りを r3 に返す。このサブルーチンを利用し、0 番地に格納されている数が 3 の倍数ならば、1 番地に 1 を、そうでなければ 0 を書くプログラムを POCO のアセンブラで書け。
6. POCO を、目標周期を 10ns にして論理合成した所、slack が-0.4ns になった。そこで目標周期を 12ns にした所 slack が 0.4ns になった。両者の動作周波数を求めよ。
7. 問題 6 の前者と後者のうちどちらが面積（ゲート数）が大きくなると考えられるか？またそれはなぜかを説明せよ。
8. 添付の POCO の Verilog 記述でレジスタファイルの書き込みを制御する信号の名称を書け。この信号はどのような命令で 1 にするかを説明せよ。
9. 256Kword の主記憶に対して 32Kword のキャッシュを設ける。ブロックサイズを 64word とした時の (1) ダイレクトマップキャッシュのディレクトリ構成、(2) 2-way セットアソシアティブキャッシュのディレクトリ構成を述べよ。
10. 最近のプロセッサのキャッシュが階層構成になっている理由を簡単に説明せよ。

全 10 点 A) POCO の命令コード

```
MV  rd,rs    rd <- rs                        00000 ddd sss 00001
AND rd,rs    rd <- rd AND rs                 00000 ddd sss 00010
OR  rd,rs    rd <- rd OR rs                  00000 ddd sss 00011
SL  rd       rd <- rd<<1                     00000 ddd --- 00100
SR  rd       rd <- rd>>1                     00000 ddd --- 00101
ADD rd,rs    rd <- rd + rs                   00000 ddd sss 00110
SUB rd,rs    rd <- rd - rs                   00000 ddd sss 00111
ST rs, (ra)     rs -> (ra)                   00000 sss aaa 01000
LD rd, (ra)     rd <- (ra)                   00000 ddd aaa 01001
LDI rd,#X       rd <- X (符号拡張)           01000 ddd XXXXXXXX
LDIU rd,#X      rd <- X (符号拡張なし)       01001 ddd XXXXXXXX
ADDI rd,#X      rd <- rd + X (符号拡張)      01100 ddd XXXXXXXX
ADDIU rd,#X     rd <- rd + X (符号拡張なし)  01101 ddd XXXXXXXX
LDHI rd,#X      rd <- X|0                    01010 ddd XXXXXXXX
BEZ rd, X   if (rd==0) pc <- pc + X          10000 ddd XXXXXXXX
BNZ rd, X   if (rd!=0) pc <- pc + X          10001 ddd XXXXXXXX
BPL rd, X   if (rd>=0) pc <- pc + X          10010 ddd XXXXXXXX
BMI rd, X   if (rd<0) pc <- pc + X           10011 ddd XXXXXXXX
JMP X       pc <- PC + X                     10100 XXXXXXXXXXX
JAL X       r7 <- pc, pc<- pc + X            10101 XXXXXXXXXXX
JR  rd                pc <- rd               00000 ddd --- 01010
```

B) POCO の Verilog 記述

```
‘include "def.h"
module poco(
input clk, rst_n,   input [‘DATA_W-1:0] idatain,
input [‘DATA_W-1:0] ddatain,   output [‘DATA_W-1:0] iaddr, daddr,
output [‘DATA_W-1:0] ddataout,output we);
reg [‘DATA_W-1:0] pc;
wire [‘DATA_W-1:0] rf_a, rf_b, rf_c, alu_b, alu_y;;
wire [‘OPCODE_W-1:0] opcode, func;
wire [‘REG_W-1:0] rs, rd;
wire [‘SEL_W-1:0] com;
wire [‘IMM_W-1:0] imm;
wire rwe, st_op, bez_op, bnz_op, addi_op, ld_op, alu_op;
wire ldi_op, ldiu_op, addiu_op;
assign ddataout = rf_a;
assign iaddr = pc;
assign daddr = rf_b;
assign {opcode, rd, rs, func} = idatain;
assign imm = idatain[‘IMM_W-1:0];
// Decorder
assign st_op = (opcode == ‘OP_REG) & (func == ‘F_ST);
assign ld_op = (opcode == ‘OP_REG) & (func == ‘F_LD);
assign alu_op = (opcode == ‘OP_REG) & (func[4:3] == 2’b00);
assign ldi_op = (opcode == ‘OP_LDI);
assign ldiu_op = (opcode == ‘OP_LDIU);
assign addi_op = (opcode == ‘OP_ADDI);
assign addiu_op = (opcode == ‘OP_ADDIU);
assign bez_op = (opcode == ‘OP_BEZ);
assign bnz_op = (opcode == ‘OP_BNZ);
assign we = st_op;

assign alu_b = (addi_op | ldi_op) ? {{8{imm[7]}},imm} :
               (addiu_op | ldiu_op) ? {8’b0,imm} : rf_b;
assign com = (addi_op | addiu_op ) ? ‘ALU_ADD:
               (ldi_op | ldiu_op ) ? ‘ALU_THB: func[‘SEL_W-1:0];
assign rf_c = ld_op  ? ddatain : alu_y;
assign rwe = ld_op  | alu_op | ldi_op | ldiu_op | addi_op | addiu_op  ;

alu alu_1(.a(rf_a), .b(alu_b), .s(com), .y(alu_y));
rfile rfile_1(.clk(clk), .a(rf_a), .aadr(rd), .b(rf_b), .badr(rs),
        .c(rf_c), .cadr(rd), .we(rwe));

always @(posedge clk or negedge rst_n)
begin
  if(!rst_n) pc <= 0;
  else if ((bez_op & rf_a == 16’b0 ) | (bnz_op & rf_a != 16’b0))
   pc <= pc +{{8{imm[7]}},imm}+1 ;
  else
   pc <= pc+1;
end
endmodule
```