

問 題 用 紙

2026	科目名	情報基礎（計算機システム）	1 / 3	通し番号
------	-----	---------------	-------	------

問題 1 次の(1)~(6)に答えよ。

- (1) 二進数で表された数値 10101b を十進数で表せ。
- (2) 次の数を 5 ビットの 2 の補数表現で表せ。表せる範囲にない場合は「不可」と答えよ。
 (a) -16 (b) 16
- (3) 100 個のデータを、その各々に 8 ビットの相異なる ID を割り当てて管理している。このまま 8 ビットの ID を使い続ける場合、最大であと何個のデータを追加しても相異なる ID を割り当てられるか答えよ。
- (4) 十進数で表された数値 67 を、以下の仕様の浮動小数点数表現により表現する。丸めを切り捨てで行うとき、その指数部と仮数部を答えよ。また、その誤差の絶対値を十進数で答えよ。

浮動小数点数表現の仕様：

符号部 …… 1 ビット（符号ありを 1、符号なしを 0）

指数部 …… 4 ビットの 2 の補数表現

仮数部 …… 5 ビット（けち表現を用い、整数部の 1 を記憶しない）

- (5) 上記(4)の浮動小数点数表現を用いて表した 2 つの数の加算を行う。(4)の数値に加算した際に**情報落ち**が発生する正の数値で、(4)の浮動小数点数表現でアンダーフローとならないものを一つ、十進数で答えよ。なお、加算結果の丸めにも切り捨てを用いる。
- (6) 図 1 の論理回路の真理値表を完成させよ。

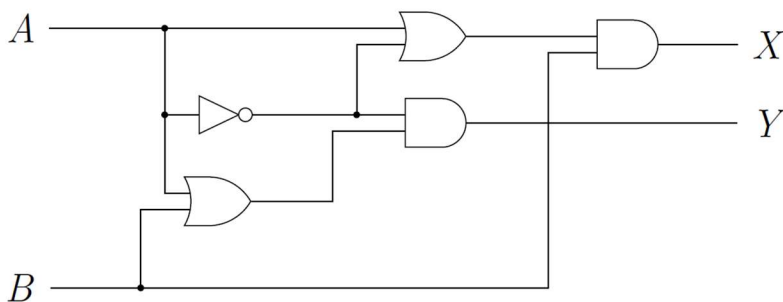


図 1：論理回路

問 題 用 紙

2026	科目名	情報基礎（計算機システム）	2 / 3	通し番号
------	-----	---------------	-------	------

問題 2 次の 32 ビットの MIPS のアセンブリ言語のプログラムについて、以下の問に答えよ。
次のページの表 1 に MIPS の命令表を、表 2 にレジスタ表を示している。

```

1.    MAIN: addi  $s0, $zero, 5        # $s0 = n = 5
2.           addi  $s1, $zero, 0      # $s1 = 0
3.           addi  $s2, $zero, 1      # $s2 = 1
4.    LOOP: add   $s1, $s1, $s2
5.           _____              # $s2 ← $s2 + 1
6.           slt   $t0, $s0, $s2
7.           beq   $t0, $zero, LOOP
8.    L1:   sw    $s1, 8($fp)
    
```

- (1) 1行目の命令を機械語のコードに変換せよ。
- (2) 7行目の命令を機械語のコードに変換せよ。
- (3) 以下のMIPSのアセンブリ言語の命令の内、レジスタ\$s1の値の下位3ビットに1を立てる命令として正しいのはどれか。次の(A)から(D)までの選択肢の中から一つ選べ。

(A) andi \$s1, \$s1, 3 (B) andi \$s1, \$s1, 7 (C) ori \$s1, \$s1, 3 (D) ori \$s1, \$s1, 7
--

- (4) 8行目の下線部のオペランドのアドレッシングモードを、以下の(a)から(e)までの選択肢の中から一つ選べ。

(a) レジスタアドレッシング (b) ベースアドレッシング (c) 即値アドレッシング (d) PC 相対アドレッシング (e) 擬似直接アドレッシング

- (5) 8行目の命令実行時に、フレームポインタ\$fpの値が十六進数で0x1001000Cであったとする。このときの実効アドレスは何か。十六進数で答えよ。
- (6) 5行目の1行を使用して、レジスタ\$s2に格納されている値に1を加算してレジスタ\$s2にその加算結果の値を格納したい。1つの命令をアセンブリ言語で書け。
- (7) 5行目の命令は何回実行されるか答えよ。
- (8) 8行目の実行によってメモリに格納される値を十進数で答えよ。
- (9) このプログラムは何をするプログラムか答えなさい。8行目でどのような値がメモリに格納されるかを、1行目でレジスタ\$s0に設定される値であるnを用いて答えよ。

問題用紙

2026	科目名	情報基礎（計算機システム）	3 / 3	通し番号	
------	-----	---------------	-------	------	--

表 1：MIPS 命令表

ニーモニック	動作	コード			
add $\$rd, \$rs, \$rt$	$\$rd \leftarrow \$rs + \$rt$	000000ss	sssttttt	dddd000	00100000
addi $\$rd, \rs, C	$\$rd \leftarrow \$rs + C$	001000ss	sssddddd	cccccccc	cccccccc
slt $\$rd, \$rs, \$rt$	$\$rs < \rt ならば $\$rd \leftarrow 1$, そうでなければ $\$rd \leftarrow 0$	000000ss	sssttttt	dddd000	00101010
andi $\$rd, \rs, C	$\$rd \leftarrow \$rs \& C$	001100ss	sssddddd	cccccccc	cccccccc
ori $\$rd, \rs, C	$\$rd \leftarrow \$rs C$	001101ss	sssddddd	cccccccc	cccccccc
sw $\$rs, C(\$rb)$	$\$rs \rightarrow M[\$rb + C]$	101011bb	bbbsssss	cccccccc	cccccccc
beq $\$rs, \rt, C	$\$rs == \rt ならば "この beq 命令 の行番号+1+C" 行 目に飛ぶ	000100ss	sssttttt	cccccccc	cccccccc

(注釈) コード中の sssss, ttttt, ddddd, bbbbbb は、それぞれニーモニック中の $\$rs, \$rt, \$rd, \rb のレジスタ番号 (表 2 参照)

コード中の cccc...ccc は、ニーモニック中の即値等の定数 C の 2 進表記 (負の数の場合は 2 の補数表現)

表 2：レジスタ表

名称	番号 (2 進)	用途
$\$zero$	00000	常に 0
$\$t0 \sim \$t7$	01000 ~ 01111	一時変数 (セーブされることを想定しない)
$\$s0 \sim \$s7$	10000 ~ 10111	一時変数 (セーブされることを想定)
$\$fp$	11110	フレームポインタ