

令和8年度専攻科入学選抜前期学力検査問題

電気情報システム工学専攻 情報系 専門 I (論理回路・計算機工学)

(1/5)

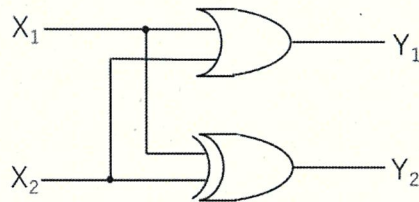
受験番号	氏 名

得 点
18

総 得 点
100

問1. 排他的論理和ゲート回路 (XOR回路) は $\overline{X_1} \cdot X_2 + X_1 \cdot \overline{X_2}$ と表せる. XOR回路と論理和ゲート回路 (OR回路) を用いて構成した下図の分離回路について, 以下問いに答えよ.

(6点×3=18点)



(1) 入力 X_1 , X_2 に対する出力を Y_1 , Y_2 とするときの真理値表を完成させよ.

なお, $\Rightarrow$ はOR回路, $\Rightarrow$ はXOR回路を表す.

(2) 否定論理積ゲート回路 (NAND回路) $\Rightarrow$ を用いて, 上図の分離回路を構成することを考える. 否定ゲート回路 (NOT回路) $\Rightarrow$ は, と表現できることを利用して, $F = \overline{X_1} \cdot X_2$ の回路図をNAND回路のみを使って構成せよ.

(3) 否定論理積ゲート回路 (NAND) のみを用いて, 上図の分離回路を構成せよ.

(1)	完答 <table border="1"> <thead> <tr> <th>X_1</th><th>X_2</th><th>Y_1</th><th>Y_2</th></tr> </thead> <tbody> <tr> <td>0</td><td>0</td><td>0</td><td>0</td></tr> <tr> <td>0</td><td>1</td><td>1</td><td>1</td></tr> <tr> <td>1</td><td>0</td><td>1</td><td>1</td></tr> <tr> <td>1</td><td>1</td><td>1</td><td>0</td></tr> </tbody> </table>	X_1	X_2	Y_1	Y_2	0	0	0	0	0	1	1	1	1	0	1	1	1	1	1	0	(2)	
X_1	X_2	Y_1	Y_2																				
0	0	0	0																				
0	1	1	1																				
1	0	1	1																				
1	1	1	0																				
(3)		(2)(3) 補足 ※設問に沿わない (NAND 回路以外を使っている) 場合は不正解とする. ※模範解答と同等の出力をする回路は正解とするが, 複雑な (例えば NAND 回路が模範解答より多い) 場合は減点 (−2点) する.																					

受験番号	氏 名

得 点
16

問2. 下記は、符号なし2進数で表現された固定小数点数の除算法である引き放し法(非回復型除算法)の手順を示している。空欄に適する数値を、解答欄右下に示すbit数で解答せよ。なお、下図では演算方法を示す+や-の記号は省いている。また、演算によって発生したエンドキャリ(最上位ビットからの桁上げ)は省いて解答すること。(2点×8=16点)

$$(106)_{10} \div (5)_{10} = (21)_{10} \quad \text{剰余}(1)_{10}$$

$$\text{被除数 } DD = (106)_{10} = (01101010)_2$$

$$\text{除数 } DS = (5)_{10} = (0101)_2$$

<手順補足>

- 被除数 DD は、部分剰余 PR (残りの下位ビット DD') と記載し、PR と DS で演算 (加算または減算) を行うことで、部分商 PQ を求める。
- 符号 S の初期値は1。演算結果の最上位ビットが、0の場合はSを1とし、1の場合はSを0とする。
- 演算結果の最上位ビットが、0の場合はPQを1とし、1の場合はPQを0とする。
- 符号 S が、1の場合は次の演算は減算を行い、0の場合は次の演算は加算を行う。

DS	符号	PR	DD'	PQ
0101	S←1	0110	(1010)	
		0101 (-		
	S←1 (1)	0001	(1010)	(2) 1
		← 1bit シフト		
	(3)	0011	(010)	
		0101 (-		
	S←0 (4)	1110	(010)	0
		← 1bit シフト		
		1100	(10)	
		0101 (+		
	S←1 (5)	0001	(10)	1
		← 1bit シフト		
		0011	(0)	
		0101 (-		
	S←0 (6)	1110	(0)	(7) 0
		← 1bit シフト		
		1100		
		0101 (+		
	S←1 (8)	0001		1

従って、 $Q = (10101)_2$, $R = (0001)_2$,

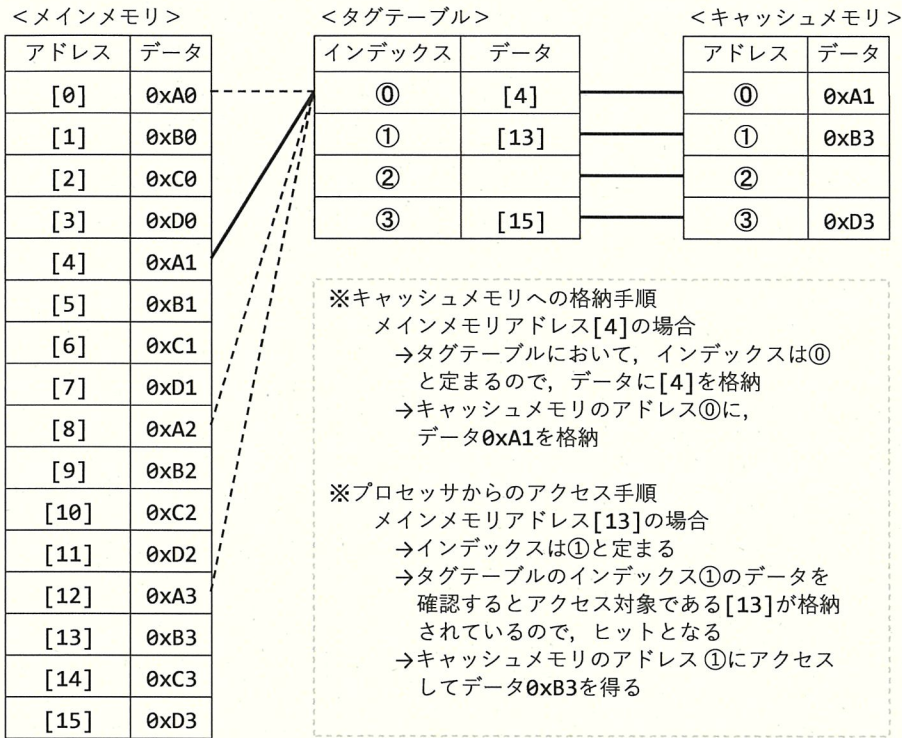
※(3)等を011(010)【bit数が少ない】や、00011(010)【bit数が多い】の様に解答した場合、2進数としての値が等しければ正解とする。ただし、PQは1bitである必要があるので、(2)(7)を01の様に2bitとした場合は不正解とする。

※注：例えば、ビット数の指定が4(2)の場合は、0000(00)の様に解答すること。

(1)	0001(1010) 4(4)	(2)	1 1	(3)	0011(010) 4(3)	(4)	1110(010) 4(3)
(5)	0001(10) 4(2)	(6)	1110(0) 4(1)	(7)	0 1	(8)	0001 4

受験番号	氏 名	得 点
		18

問 3. 下図はダイレクトマップ方式のキャッシュメモリを簡易的に示した概念図である。下図の状態を「初期状態」とする。タグテーブルのデータが空欄の箇所は、キャッシュメモリにデータが格納されていないことを意味する。図中の「※キャッシュメモリへの格納手順」と「※プロセッサからのアクセス手順」を参考に、下記問いに答えよ。(3点×6＝18点)



- ・アドレス[15]にアクセスするとヒットするので、キャッシュメモリのアドレス (1) にアクセスして目的のデータを得る。①～③から選んで答えよ。③ (15 mod 4 = 3 なので)
- ・アドレス[5]にアクセスするとミスするので、ブロック置換を行うが、その際に格納する場所は、タグテーブルのインデックス (2) に格納する。①～③から選んで答えよ。① (5 mod 4 = 1 なので)
- ・アドレスを格納する場所(インデックス)は計算式で定められる。その計算式を、以下の空欄を埋めるか、あるいは自由記述により答えよ。4で割った余り。インデックス＝メインメモリのアドレス mod 4 など
「メインメモリのアドレスを (3) をインデックスとする」
- ・初期状態からアドレス[8]→[8]と順にアクセスした場合、この2回のアクセスにおけるキャッシュヒット率は 50 %となる。また、初期状態からアドレス[4]→[15]→[12]→[4]と順にアクセスした場合、この4回のアクセスにおけるキャッシュヒット率は (4) %となる。空欄にあてはまる数値を答えよ。
[4]ヒット→[15]ヒット→[12]ミス→[4]ミス (12を格納したのでミス)
- ・初期状態からアドレス[5]→[9]→[5]→[9]と順にアクセスした場合、この4回のアクセスにおけるキャッシュヒット率は (5) %となる。一方、マッピング方式がダイレクトマップではなくフルアソシティブ方式だった場合は、初期状態から同様の順番のアクセスをした場合のキャッシュヒット率は最大で (6) %である。空欄にあてはまる数値を答えよ。(5)同じ場所にマッピングされるので、毎回ミスとなる。/
(6) [5]ミス→[9]ミス→[5]ヒット→[9]ヒット。最初の2回は必ずミス。格納場所は自由なので、[5]と[9]を残すように格納すれば、残りの2回はヒットにできる。

(1)	③	(2)	①	(3)	4で割った余り, インデックス＝メインメモリのアドレス mod 4
(4)	50 [%]	(5)	0 [%]	(6)	50 [%]

受験番号	氏 名

得 点
36

問4. 下記の文の空欄に最も適する用語や数値を、解答欄に記入せよ。 (3点×12 = 36点)

- AND や OR などの組み合わせ回路と異なり、過去の入力によって決まった状態と現在の入力によって出力が決まる順序回路の一種を、カタカナ用語で (1) という。その性質から記憶回路として利用される。フリップフロップ
- コンピュータは2進数を用いている。「2進」は英単語で (2) と書く。binary
- 近年、スマート家電やウェアラブル端末の普及により「いつでも」「どこでも」コンピュータを利用できる時代が到来している。「いつでも」「どこでも」という状態を表す言葉として (3) という言葉がある。ユビキタス
- ネットワークで通信を行うときの通信規約のことを (4) という。代表的な例としては、HTTP や TCP/IP などがある。プロトコル
- オペレーティングシステムの中核となるソフトウェアのことを (5) という。これはオペレーティングシステムの最も重要で基本的な部分であり、ハードウェアの管理等を行う。カーネル
- Python は、実行時にコンパイルを必要とせず、プログラムを1文ずつ解釈しながら実行するため、 (6) 型のプログラミング言語といえる。インタプリタ
- コンピュータにおける命令実行サイクルが、「命令フェッチ→デコード→実行→結果格納」の4ステージの場合を考える。1個の命令を実行するときに必要なステージ数は4ステージであるが、命令パイプライン処理を用いて、100個の命令を実行するときに必要なステージ数は (7) である。
 $S_p = I + D - 1 = 100 + 4 - 1 = 103$
- コンピュータでは、ひとつのプロセスの中で複数のスレッドを並行して実行することで、命令実行の多重化を図る。この複数のスレッドによる並列処理をカタカナ用語で (8) という。
↑マルチスレッド ↓マスク
- コンピュータの割り込み機構において、割り込みの有無を示すレジスタは「割り込みフラグレジスタ」という。また、割り込みの有効無効を示すレジスタは「割り込み (9) レジスタ」という。
- DRAM (Dynamic Random Access Memory) 等の半導体メモリには、コンデンサからの漏れ電荷を補うために定期的に電源を供給しないと格納内容が消失する性質のものがある。半導体メモリにおけるこの性質を「 (10) 性」という。揮発
- RAID1 とは、複数のハードディスクに全く同じデータを格納することで、ハードディスクが1台壊れても、他のハードディスク内にデータが残っていることを利用した、信頼性向上のための技術である。RAID1はこの構造から、カタカナ用語で (11) と呼ばれる。ミラーリング
- 複数のプロセスが、互いに資源の解放を待ち合わせる状態になり、すべてのプロセスが実行できなくなる現象のことを (12) という。デッドロック

※長音 (ー) の有無による表記揺れは正解とする (例: (6)インタープリターも正解)。ただし、誤った表記は不正解 (例: (5)カネル)
※濁音半濁音の誤りは-1点 ※漢字をひらがな、カタカナで書いてもよい。ただし、漢字の誤りは不正解とする。

(1)	フリップフロップ	(2)	binary	(3)	ユビキタス	(4)	プロトコル
(5)	カーネル	(6)	インタプリタ	(7)	103	(8)	マルチスレッド
(9)	マスク	(10)	揮発(性)	(11)	ミラーリング	(12)	デッドロック

受験番号	氏 名

得 点
12

問5. 下記の問いについて、最も適切な選択肢（ア～エ）を選び、解答欄に記入せよ。

(3点×4＝12点)

(1) 「トランジスタ」についての説明文として、正しいものを選び。

- ア. バイポーラ型トランジスタは、現在、CPUやメモリの主要部品として使用されている。CMOS
- イ. CMOSトランジスタは、大電流・大電圧性に優れている。高速・省エネルギー性
- ウ. ユニポーラ型トランジスタは、電子と正孔の両方がキャリアとして寄与する。どちらか
- エ. SiCトランジスタは、高温・高圧での動作に優れている。●

(2) コンピュータにおける命令語の形式である「2アドレス形式」についての説明文として、正しいものを選び。

- ア. 命令の実行前と実行後で、オペランドで指定されたレジスタの格納内容が変化する。●
- イ. 加算などの2項算術演算の命令の実行に、スタックを必要とする。0アドレス形式
- ウ. 3アドレス形式と比較して、命令語長が長い。短い
- エ. デスティネーションオペランドが2つの命令である。ソース1，ソース&デストが1

(3) コンピュータ性能の評価指標である「TPC」についての説明文として、正しいものを選び。

- ア. 1つの命令を実行するときに必要な時間で示される指標である。TPI
- イ. Time Per Clock cycle の略である。●
- ウ. 「平均命令実行クロックサイクル数」を意味する指標である。マシンサイクル時間
- エ. 他の性能指標を利用すると、 $TPI \times CPI$ という計算で求められる。 $TPI \div CPI$

(4) 入出力制御における「入出力命令」か「メモリマップ入出力」のいずれかの説明文を以下に示す。「メモリマップ入出力」の説明文として適切なものを選び。

- ア. 入出力装置を制御する場合、入出力装置専用のマシン命令を使用する。入出力命令の説明
- イ. 入出力制御のために、メインメモリ空間を使用しない。メモリマップ入出力は使用するので不適
- ウ. 入出力装置を制御する場合、メモリへのread/write命令を使用する。●
- エ. 接続する入出力装置があらかじめ定まっている場合に有効である。入出力命令の説明

(1)	エ	(2)	ア	(3)	イ	(4)	ウ
-----	---	-----	---	-----	---	-----	---