

令和8年度
専攻科入学者学力選抜検査問題

(専 門 科 目)

生産システム工学専攻 C群

電気電子工学, 計算機工学, 計算機システム,
計算機科学, ソフトウェア

受験番号	
------	--

(注 意)

- 1 指示があるまで開かないでください。
- 2 問題は1ページから 8ページまであります。検査開始の合図のあとで確認してください。
- 3 貸与する電卓を使用しても構いません。
- 4 問題は5問です。その中から3問を選択して解答してください。下の表に、選択した問題番号に丸 (○) 印をつけてください。なお、選択した問題以外に解答しても採点されません。

問題番号	1	2	3	4	5
選択した番号					

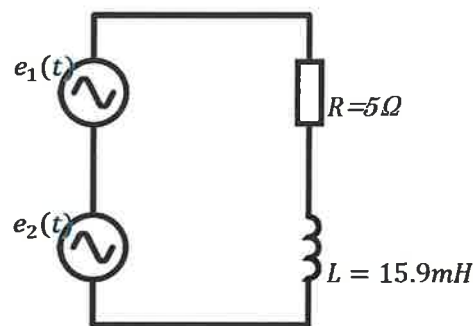
独立行政法人国立高等専門学校機構

福井工業高等専門学校

1 以下の電気電子工学に関する問題に答えなさい。

右図に示す抵抗とコイルの直列回路において、周波数 $50[\text{Hz}]$ の2つの電圧源 $e_1(t) = 50\sin(\omega t + \frac{\pi}{2})$ 及び $e_2(t) = 50\sin(\omega t + \frac{\pi}{6})$ を加えた。

問 1. それぞれの電圧 $e_1(t)$ 及び $e_2(t)$ をオイラー形式で表示しなさい。

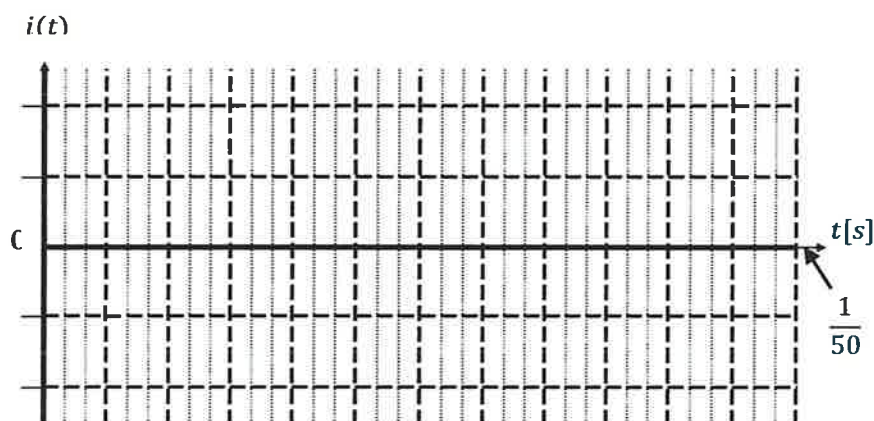


$$\dot{E}_1 = \quad , \quad \dot{E}_2 =$$

問 2. 回路全体の合成インピーダンス $\dot{Z}$ をオイラー形式で求めなさい。ただし、 $\pi \times 15.9 = 50$ として計算に用いなさい。

$$\dot{Z} =$$

問 3. 回路全体を流れる電流 $i(t)$ を瞬時値形式で求めて波形を描画しなさい。ただし、波形を描画する際に正弦波が横軸と交わる点、最小値および最大値となる点、有名角 ($\frac{\pi}{6}[\text{rad}]$ や $\frac{\pi}{3}[\text{rad}]$ に関連した) 点をプロットしなさい。また縦軸には最大値及び最小値、横軸にはグラフが横軸と交わる時刻を記入しなさい。



2 以下 の計算機工学に関する問題に答えなさい。

問 1. n 桁 2 進数の演算を行う方式には、直列加算器と並列加算器があります。直列加算器は 2 進数の最下位桁 (LSB) から最上位桁 (MSB) に向かって 1 桁ずつ順に加算していきます。それに対して、並列加算器は 2 進数の各桁の加算を同時に行います。

では、RCA (リプルキャリー加算器) は上記のどちらの加算器にあたるか答えなさい。

解答欄: _____

問 2. RCA と CLA (キャリールックアヘッド加算器) を比較して多 bit の加算に向いているのはどちらか、またその理由を具体的に答えなさい。

解答欄: 加算器名 _____

理由 _____

問 3. 図 1 は下位桁のキャリーアウトを D-FF に保存する直列加算器の例です。FF の現在の状態を Q 、次の状態を Q^+ として、状態遷移表を作成しなさい。

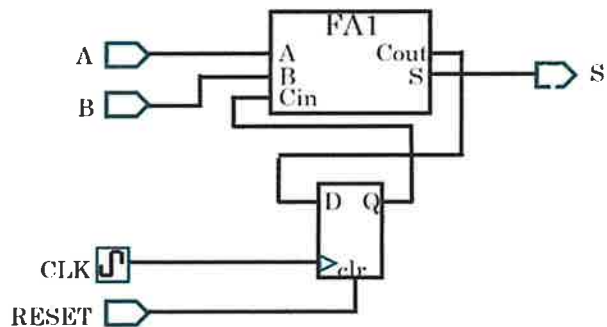
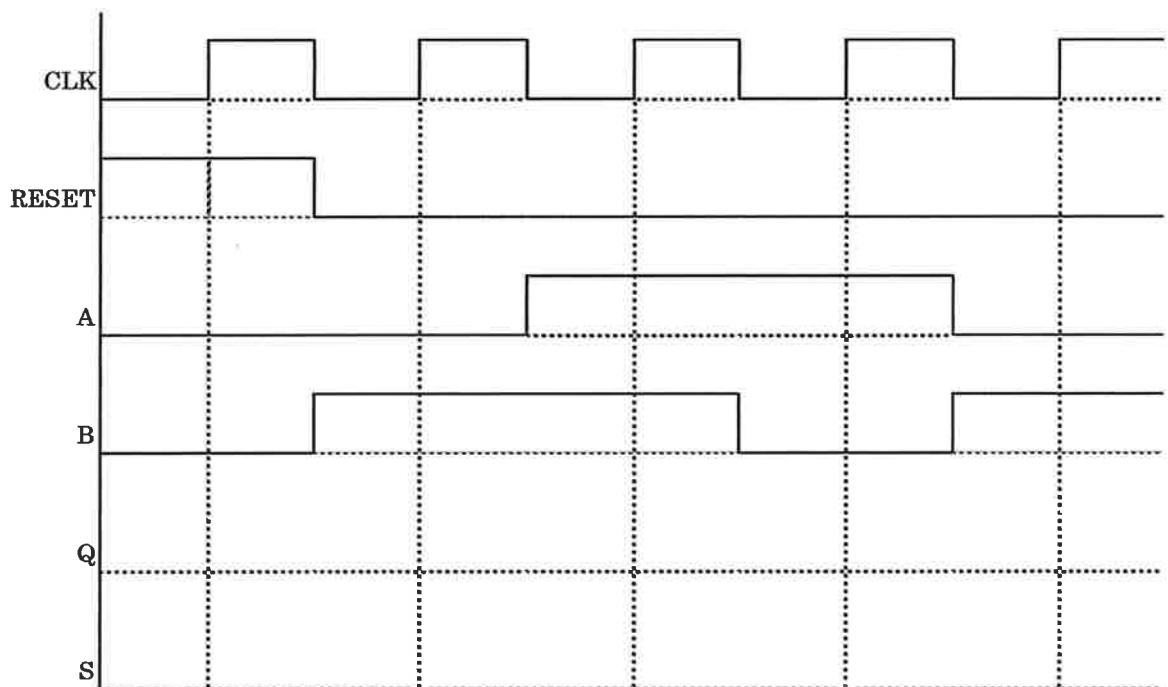


図 1 直列加算器

問4. 図1の直列加算器の状態遷移図 (Mealy 図) を作成しなさい。

問5. 図1の回路で4bit 同士の加算, $0110 + 1011$ を行った時の下記のタイミングチャートを完成させなさい。
ただし, FF の遅延時間は無視して良いとします。



3 以下の計算機システムに関する問題について答えなさい。

問1. 以下のア～カの割込みの中で、内部割込みに分類されるものを全て選んで解答欄に記入しなさい。

- ア 瞬時停電などの電源異常による割込み
- イ ゼロで除算を実行したことによる割込み
- ウ 入出力が完了したことによる割込み
- エ メモリパリティエラーが発生したことによる割込み
- オ 未定義命令実行によるプログラム割込み
- カ ユーザプログラムから OS のサービス機能呼び出す命令(カーネル呼び出し)の実行で発生するシステムコール(カーネル呼び出し, SVC)割込み

解答欄: _____

問2. ページング方式の仮想記憶におけるページ置換えアルゴリズムとして LRU 方式を採用します。主記憶に割り当てられるページ枠が4のとき、ページ1, 2, 3, 4, 5, 2, 1, 3, 2, 6の順にアクセスすると、ページ6をアクセスする時点で置き換えられるページの番号を答えなさい。ここで、初期状態では主記憶にどのページも存在しないものとします。

解答欄: _____

問 3. 絶対パス名 /a/a/b/c を持つディレクトリがカレントディレクトリであるとき、
相対パス ../../a/b/file をもつファイルを絶対パス名で表現しなさい。

ここで、ディレクトリ及びファイルの指定方法は、次の規則に従うものとします。

[ディレクトリ及びファイルの指定方法]

- ① ファイルは、"ディレクトリ名/.../ディレクトリ名/ファイル名"のように、経路上のディレクトリを順に"/"で区切って並べた後に"/"とファイル名を指定します。
- ② カレントディレクトリは"."で表します。
- ③ 1階層上のディレクトリは".."で表します。
- ④ 始まりが"/"のときは、左端にルートディレクトリが省略されているものとします。
- ⑤ 始まりが"/", ".", ".."のいずれでもないときは、左端にカレントディレクトリ配下であることを示す"/"が省略されているものとします。

解答欄: _____

4 以下の計算機科学に関する問題に答えなさい。

問 1. 図 1 に示す状態遷移特性をもつ 3 元マルコフ情報源を考えます。状態は S_0 , S_1 および S_2 です。 S_x に遷移するとき情報源は x を出力します(x は 0 から 2 をとります)。遷移確率は各矢印弧に示されています。

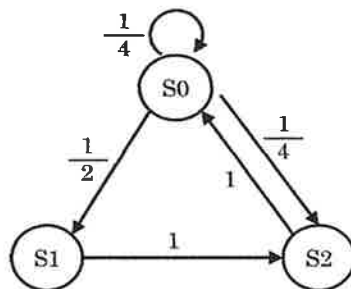


図 1 状態遷移図

(1) P_x を S_x に留まる定常確率とします。 P_0 , P_1 および P_2 を既約分数で答えなさい。

P_0		P_1		P_2	
-------	--	-------	--	-------	--

(2) 図 1 のマルコフ情報源のエントロピー H を既約分数で答えなさい。

H	
-----	--

5 以下のソフトウェアに関する問題に答えなさい。

問1. 以下のJava による再帰プログラムについて設問に答えなさい。

<pre>public class Main { public static int mul_rec(int x, int y, int ans) { if (y == 0) { return ans ; } else { if (y % 2 == 1) ans = ans + x ; return mul_rec(x * 2 , y / 2 , ans) ; } } public static int mul_loop(int x, int y, int ans) { while(y > 0) { if (y % 2 == 1) ans = C ; x = D ; y = E ; } return ans ; } }</pre>	<pre>public static void main(String[]args) { System.out.println(/* (A) */ mul_rec(10 , 5 , 0)) ; System.out.println(/* (B) */ mul_rec(10 , 100 , 0)) ; /* (A)と同じ結果が表示されること */ System.out.println(mul_loop(10 , 5 , 0)) ; } /* main() */ } /* class Main */</pre>
--	---

(ア) 上記のプログラムで `main()` を実行した場合の、`mul_rec()` を呼び出した処理 (A),(B) の実行結果を答えなさい。

解答欄 (A) _____ (B) _____

(イ) `mul_rec()` と同じ結果が得られる、ループ処理による関数 `mul_loop()` を作成したい。

, , に相応しい処理を答えなさい。

解答欄 _____ _____ _____

(ウ) 正の自然数 N, M において `mul_rec( $N, M, 0$ )` を実行した時の処理に要する時間をオーダー記法で示すと何がふさわしいか、以下の選択肢の中から適切なものを選びなさい。

選択肢 $O(N)$, $O(M)$, $O(NM)$, $O(\log N)$, $O(\log M)$, $O(\log N \log M)$

解答欄 _____

(エ) `mul_rec()` のようにループ処理に置き換え可能な再帰プログラムは、一般的に何と呼ばれるか答えなさい。

解答欄 _____

問2. 以下のJava による行列式の乗算を行うプログラムについて設問に答えなさい。

```
public class Main {
    public static void main( String[] args ) {
        int[][] a = {
            { 1 , 2 , 3 } ,
            { 4 , 5 , 6 } ,
        };
        int[][] b = {
            { 2 , 1 } ,
            { 4 , 3 } ,
            { 6 , 5 } ,
        };
        if ( a[0].length ==  ) {
            int[][] c = new int[  ][ b[0].length ] ;
            for( int ay = 0 ; ay < a.length ; ay++ ) {
                for( int bx = 0 ; bx <  ; bx++ ) {
                    int sum = 0 ;
                    for( int ax = 0 ;  ; ax++ )
                        sum = sum + a[ay][ax] *  ;
                     = sum ;
                }
            }
            for( int cy = 0 ; cy < c.length ; cy++ ) {
                for( int cx = 0 ; cx < c[0].length ; cx++ )
                    System.out.print( c[cy][cx] + " " ) ;
                System.out.println() ;
            }
        }
    }
}
```

$$\begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{pmatrix} \times \begin{pmatrix} 2 & 1 \\ 4 & 3 \\ 6 & 5 \end{pmatrix} = \begin{pmatrix} 28 & 22 \\ 64 & 49 \end{pmatrix}$$

図 5-2 行列積の計算例

上記のプログラムは、図 5-2 に示す行列式のように、2 次元配列 a,b の行列積を 2 次元配列 c に格納し、結果を出力したい。2 次元配列 a,b の大きさが行列積の条件を満たさない場合は処理を行わない。

与えられる行列は 1 行 1 列以上の配列で、2 次元配列 a,b の各行の列数は同じとする。

処理 , , , , , に相応しい処理を答えなさい。

解答欄