

応用情報技術者試験 復習レポート：LEDを用いたデジタル時計の設計と制御

1. はじめに：本設問の背景と学習の意義

組み込みシステム開発において、プログラマブル入出力装置（PIO）を介したデバイス制御、ダイナミック点灯による表示の最適化、そして割込み処理を用いたリアルタイムな状態監視は、設計者が必ず習得すべき核心的な技術要素です。本設問は、時計という身近な題材を通じて、MPUがハードウェア・レジスタをどのように操作し、物理的な出力装置（LED）や入力装置（スイッチ）と対話するかを網羅的に扱っています。単なる過去問解説を超え、ハードウェアの物理的特性（チャタリング等）をソフトウェアがいかに吸収し、システムの信頼性を担保しているかという「ハード・ソフトの相互作用」を理解することは、実務におけるデバッグ能力やアーキテクチャ設計能力の向上に直結します。本レポートでは、ハードウェア構成の基礎から、ソフトウェアによる高度な制御ロジックまでを戦略的視点で分析します。

2. ハードウェア構成とPIO（プログラマブル入出力装置）の制御

本システムは、時刻管理を行うRTC、表示用の6個の7セグメントLED、ユーザー入力を受け付けるスイッチ（SW）、およびこれらを制御するPIOで構成されています。PIO制御の要諦は、各端子の方向性を定める **PIO_R（入出力制御レジスタ）** と、実際の入出力値を保持する **PIO_D（データレジスタ）** の役割分担を正確に把握することにあります。

設問1の分析：PIO_R設定値「3FFF」の導出プロセス

PIO_Rの各ビット R_i は、対応する D_i を出力（1）にするか入力（0）にするかを決定します。ソースコンテキストに基づき、各ビットの用途と設定値を判定します。**ビット割り当てと入出力判定表**

ビット	R15	R14	R13	R12	R11	R10	R9	R8	R7	R6	R5	R4	R3	R2	R1	R0
用途	未使用	SW	LED5	LED4	LED3	LED2	LED1	LED0	dp	g	f	e	d	c	b	a
入出力	入力	入力	出力	出力	出力	出力	出力	出力	出力	出力	出力	出力	出力	出力	出力	出力
設定値	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1

【16進数への変換プロセス】

- 2進数表記：0011 1111 1111 1111
- 4ビットごとの変換：0011→3、1111→F、1111→F、1111→F
- 結果：3FFF💡 **講師のワンポイント・アドバイス：ケアレスミスを防ぐ計算術** 試験本番で最も多いミスは、ビットの数え間違いによる変換ミスです。必ずLSB（ R_0 ）側から4ビットずつ区切り、区切り線を入れてから変換する習慣をつけてください。また、 R_{15} （未使用）と R_{14} （SW）を「入力(0)」と判定できるかが、正解への分岐点となります。「So What?」**レイヤー：設計上のフェールセーフ** 実務において、未使用ビットを明示的に「入力」に設定することは、物理的な**電氣的衝突（コンテンション）を防止する**ための重要な安全設計です。もし未使用端子を

出力設定にしてしまい、基板上で予期せず接地（GND）や電源（Vcc）に接続されていた場合、過電流による素子の破壊を招く恐れがあります。

3. ダイナミック点灯方式と表示制御アルゴリズム

本装置は、視覚の残像現象を利用した「ダイナミック点灯方式」を採用しています。これは1ミリ秒ごとの割込みに合わせて点灯対象のLEDを高速に切り替える手法で、消費電力抑制と配線リソースの節約を両立させています。

設問2の分析：流れ図におけるロジックの解決

- **表示データの選択（空欄 a, b）** $swState = 0$ （スイッチが押されていない）時は「時分秒」を表示するため、配列 **Time** を選択します。逆に $swState = 1$ （No側）の時は「年月日」を表示するため、配列 **Date** を選択するのが妥当です。
- **ビット位置の算出（空欄 c）** カウンタ i （0～5）と点灯させるLEDのビット（D13～D8）の関係を式にします。 $i = 0$ のとき13ビット目、 $i = 5$ のとき8ビット目となるため、 $13 - i$ が導き出されます。
- **カウンタのリセット条件（空欄 d）** 全6個のLEDを一巡させるための境界条件です。 i を加算した直後の判定であるため、最後のLED（ $i = 5$ ）の処理後に i が 6 になった瞬間にリセットする必要があります。なお、プログラムの堅牢性を高めるため、実務的には $i = 6$ だけでなく、オーバーランを考慮した $i \geq 6$ や $i > 5$ と記述するのも一般的です。

エキスパート解説：PIO_D 設定値の構造

問題文に「LED3に '4' を表示するには 0866 を書き込む」という記述があります。これを分解すると、制御の仕組みがより鮮明になります。

- **セグメントデータ（下位8ビット）**：'4' を表すパターンデータ 0066（D6, D5, D2, D1が '1'）。
- **LED選択ビット（上位8ビット）**：LED3に対応する D_{11} を立てる 0800。
- **合成結果**： $0800 \mid 0066 = 0866$ 。このように、1つのレジスタ内で「どのLEDを（上位）」かつ「どのパターンで（下位）」点灯させるかを合成して出力している点がポイントです。

4. 入力インターフェースの信頼性：チャタリング対策

物理的な「機械式スイッチ」には、接点が接触する瞬間に微細な振動が生じ、オン・オフが高速に繰り返される **チャタリング（接点の物理的な跳ね返り）** 現象が必ず発生します。

設問3の分析：時間軸による信頼性の確保

図5の①の処理では、スイッチの状態 D_{14} が変化した際、直ちに反映させずカウンタ cnt を 10 に設定しています。本システムは 1ミリ秒間隔 で割込みが発生しているため、10回連続で同じ状態が継続することを確認する、すなわち **10ミリ秒間の安定** を待って初めて $swState$ を更新しています。この対策の戦略的意義は、チャタリングを「複数回のス

イッチ押下」と誤認することを防ぎ、表示の激しい点滅や誤動作を排除することにあります。「So What?」レイヤー：ソフトウェア制御の優位性 チャタリング対策をハードウェア（コンデンサやIC）ではなくソフトウェアで行うことには、**部品コストの削減**と**基板の小型化**、そして**柔軟な調整可能性**という大きなメリットがあります。物理的な接点の経年劣化によりチャタリング時間が延びたとしても、ソフトウェア上の定数を変更するだけで対応が可能となります。

5. 総括：組み込みシステム設計における「協調設計」の重要性

本設問を通じて、ハードウェアの物理的仕様を正確に理解し、それを補完・制御するソフトウェアロジックを構築する「ハード・ソフト協調設計」の重要性を再認識することができます。プロフェッショナルな設計には、以下の視点が不可欠です。

- **レジスタレベルでの厳密なビット操作**：PIO_Rの設定に見られるように、1ビットの定義ミスがシステムの沈黙や物理的な故障を招くという緊張感を持つこと。
- **リソース活用の最適化アルゴリズム**：ダイナミック点灯のように、人間の感覚特性（残像）を利用して、限られたI/Oリソースで最大限の機能を実現する工夫。
- **物理現象に対する防御設計**：ハードウェアは「理想的な挙動をしない」という前提に立ち、チャタリング対策のようなソフトウェア的な防御壁によって信頼性のラストワンマイルを埋める姿勢。以上の学びを、今後のシステム開発および試験対策の確かな指針としてください。