

わざわざ言うまでもないと思いますが、解答や解説が合っている保証はありません。
ご容赦を。

第 1 問

1 .

(ア)

RTL 論理 レイアウト設計 マスクデータ (マスクパターン)

(イ)

与えられた論理式・算術演算等を簡単化し、ゲートのみの回路に変換する。
ゲートの幾何的配置とその間の配線経路を決定し、チップ全体の構成を確定させる。

(ウ)

レジスタ数や演算器数などが明確になったモデルを使用し、サイクルを意識した入力
を行い出力を調べる。

ネットリスト：論理素子の接続で表現されたモデルを使用し、遅延などを加えた入力
を行い出力を調べる。また、論理合成の前後で回路が等価であるかを検証する。

レイアウト設計の前後で論理が等価であるかを検証する。

(エ)

LSI テスタを用いて設計通りに動作するかを調べ、不良品を選別する。特性を調べる
DC テスト・AC テストなどに加え、可能性の高い故障モデルを作成しその故障が起き
ているかを調べることにより論理回路の故障診断を行う。

2 .

ASIC：与えられたシステムを直接ハードウェアで実現するので、その使用に限っては最
も小さいかつ高性能だが、設計には時間と費用がかかる。

FPGA：チップを作ってから実際の配線等をプログラムできる。演算素子の配置や配線パ
ターンはあらかじめプロトタイプとして用意されているので、設計自体も楽。

マイクロプロセッサ：原則的にソフトウェアプログラムで命令を逐次実行するため速度
が不足し、追加のハードウェアが必要になる。ただし汎用性は高く、安価な大量生産
も可能。

3 .

例：デジタルカメラ

撮影素子から出力されたアナログデータが A/D 変換され、画像処理プロセッサにより
色々な処理を受けて PC 等での利用が可能なデジタルデータ(JPEG など)に変換され、最
最終的にメモリーカードに保存されている。

動画撮影のための音声処理機構や、ライブラリ管理のためのファイル管理システム手
ぶれ防止のためのソフトウェアなどの機能が必要とされる。

第 2 問

1 .

「除算器」の存在しないマイクロプロセッサで剰余を計算することを考えると、減算を
適当な回数だけ実行することで求めることができるが、この減算は 1 クロックに 1 回の

逐次実行であるため、処理速度は遅くなる。

一方ハードウェアでは、複数の演算を並列に実行することで高速に剰余を求めるアルゴリズムが実装できるため、処理速度が速くなる。また、この計算に必要な素子だけを使って構成できるため、この計算に使わない素子の多いマイクロプロセッサに比べて消費電力も削減できる。

2 .

例えば $3+2$ の計算など、ソフトウェアでも 1 命令で実行できるものに関しては、ハードウェアで実行したとしても並列に実行できる演算がないため、性能向上は期待できない。

3 .

システムレベル設計は、要求仕様の作成、システム仕様の作成、システムアーキテクチャ設計の三段階からなる。まず、漠然とした顧客や企画部からの要求を定量化するのが要求仕様の作成であり、続いて要求仕様の具体的実現方法を決めるのがシステム仕様の作成である。最後に、システム実現の構造・方式を決めるのがシステムアーキテクチャ設計である。組込みシステムにおいては、この段階でハードウェアとソフトウェアの分割を行うことになる。

分割について詳しく見ると、全てをソフトウェアで作ると仮定した状態からハードウェアで実現できそうなものはハードウェアで実現することとし、適当な分割を行った上で性能・面積（コスト）・消費電力の見積りを行う。いくつかの分割パターンについてこの見積り値を比較し、最適と思われるパターンで分割を行う。

例：DVD レコーダの設計（適当。要求があった点に関してだけ）

顧客等からの要求：映像も音もすげえいいやつを作ってくれ！

要求仕様：ハイビジョン解像度 $\circ \circ \times$ ピクセル、5.1CH サポート

システム仕様：動画圧縮に H.264 を採用、5.1CH サポート DolbyAC3 を標準サポート

HW/SW 分割：Dolby をハードウェア、H.264 をソフトウェアが担当

（解説）

第 1 問

1 .

（ア）いきなりきわどい問題です。このフロー図はイマイチ納得できない...用語も微妙だし。システムレベル記述というのは、動作記述のうちハードウェアで実現する部分と考えていいと思います。すると、そこから RTL（レジスタ転送レベル）記述を作成、そこから論理合成を行いネットリスト（＝論理記述）を作成、そこからレイアウト設定を行いマスクデータを作成。マスクデータというのは、実際の基板に配線を描いたようなものです。「ネットリスト」にはいくつか意味があるようなのでやや怪しいですが、消去法で考えると多分この答えしかないかなと。参考資料のフロー図のほうが紛れがないので参照を薦めます。あと、システムレベル記述→RTL 記述には動作合成（高位合成）という操作が存在します。しかしこれはまだまだ実用化されている段階ではないので省略されているのではないかと。

ちなみにこの第 1 問は池田先生の出題だと思われます。池田先生はハードウェア、藤田先生はハードウェアと表記するようなので。

（イ） で VerilogHDL などの記述から論理回路図を構成し、 で論理回路図から実際の空間的な配置を決定している、というだけです。

(ウ) ポイントは、検証には、要求仕様を満たしているかを調べる仕様検証(プロパティチェック)と、2つのハードウェア記述の機能・論理が等価であるかを調べる等価性検証の二種類がある、ということです。仕様検証に関する解答の記述は怪しいので各自補完をお願いします。また、システムレベル記述とRTL記述の等価性検証が必要なのかはよく分かりません。動作合成を行った時には必要ははずですが、今はとりあえず省きました。

(エ) さらに怪しいのでなんとなく感じ取るだけにしてください。検証との違いは、
検証：設計が仕様を満たすか
テスト：設計通り動作するか
ということです。テストは実際に製品が完成して初めて行います。

～参考～

「B4章 機能・論理検証 1 動作合成(1)」 「B5章 機能・論理検証」
VLSI 設計工学 2004-1
池田先生の資料「ハードウェアにおける設計表現と設計フロー」
「ハードウェアの故障とテスト」

第2問

1, 2.

ソフトウェアとハードウェアの違いを簡単にまとめると、

～ソフトウェア～

- ・逐次処理で、単位時間に単位処理を実行
- ・仕様の変更が可能
- ・消費電力が大きい

～ハードウェア～

- ・全ての処理を並列に実行できるが、実行タイミングの同期が必要
- ・仕様の変更ができない
- ・消費電力が小さい

と言ったところです。特定の動作を実行するには基本的にハードウェアのほうが優れています。また、ソフトウェアは逐次処理なので命令数が多くなると速度が落ちるので、解答はこの点に触れていればいいと思います。ビット数が32を超える数の演算、などに対してもソフトウェアの速度が落ちるのは明らかです。

2問目はよく分かりません。こんな単純でいいんかいな。

3.

システムレベル設計とは、要は商品の企画段階からハードウェア・ソフトウェアの分割までです。概念は解答のような感じでいいと思いますが、具体例というのがクセモノです。私は機械に詳しくないのでシステムの仕様全てについてなんて書けません。申し訳ありませんが各自資料等を参照して適当に補完してください。説明のポイントは、分割の際にハードウェアで実現するものとソフトウェアで実現するものを最初の要求の段階で書いて、それを段階を踏んで具体化していくことです。

～参考～

池田先生の資料「01-Intro」
「A3章 SoC はどのように作られるか」「D4章 システムアーキテクチャ設計技術4」