

- 徒然なるままに
- Xtuneの作り方

あなたは

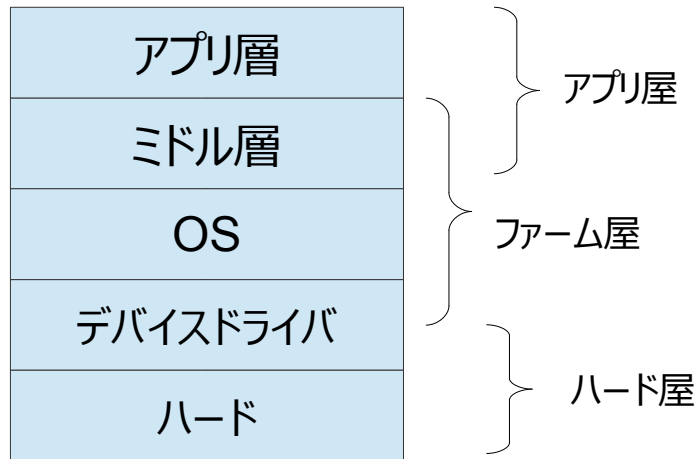
ハード屋ですか？

ファーム屋ですか？

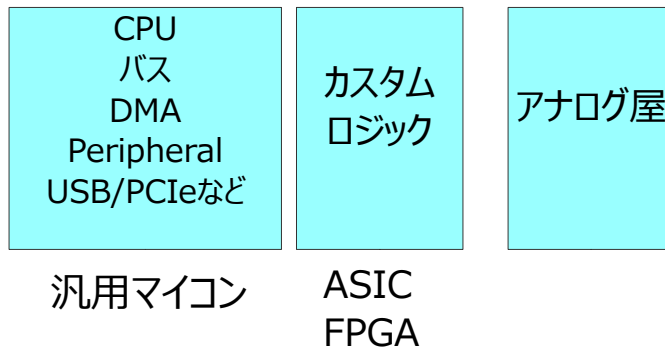
アプリ屋ですか？

「キヤノンでハード採用」だからハード屋だよ！
ってのはおいといて、少し考えてみた。

組み込みの設計のざっくりとしたイメージ



ハード例



自分のレイヤーの上下を覗いてみる気持ちが必要
→レイヤー間の狭間が面白い。

キヤノンはハード屋とソフト屋をわけて開発することが全体最適だと考えているようだ。
→ハードがわからないソフト屋、ソフトがわからないハード屋を量産して何が楽しいのか？
会社にとっては最適かも知れないが個人的にはおもしろくないじゃん。

ざっくり定義してみた

ハード屋 (デジタル)

- ・最小粒度はクロック (フリップフロップ)
- ・メモリをデバイスとしてとらえる
→アクセスイメージがある
- ・割り込みの概念を理解している
- ・バスアーキテクチャー/CPUのハード概念あり

ファーム屋

- ・最小粒度はアセンブラ
- ・メモリを箱としてとらえる
- ・割り込みを使いこなす
- ・RTOSに精通していて使いこなす

アプリ屋

- ・プログラム言語より下のレイヤーはあまり気にしない
最近の言語はメモリって何？ っていうくらい抽象化されている。
- ・割り込みなどという低レイヤーな概念は必要なく、ハンドラという言葉で隠蔽されている
- ・フレームワーク/ライブラリを使いこなす能力が必須

ハード屋からソフト屋の領域を覗く方が案外簡単じゃない？

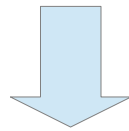
→ハードの情報よりソフトの情報の方がWebや書籍にいっぱい転がっているのは大きな強み。

→開発環境はFreeが大半

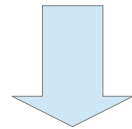
→正直電気回路の勉強よりソフトの方が楽しい

そこで

各レイヤーのスペシャリストとして勝負するのではなく、
「それなりにどのレイヤーもなんとなくできる」って人の需要もあるはずだ



「各レイヤー間は疎結合で、レイヤー間はお互いの詳細を知らなくても実装可能」
ってのが建前だが、全ての設計がきれいにできているわけではない。
広く浅くできるってこともそれなりに重要（理想は広く深くであるが）



「器用貧乏のすすめ」 という観点で考えてみる

というか私自身がスペシャリストの道は諦めて、こっちで生きる道を模索しています

「器用貧乏」になるために

- ・同じ部署で同じ仕事を長くしていると損という意識を持とう！

仕事は最初の半年～1年くらいが一番たいへんだが、自分の成長曲線も高い。
あるところから惰性で仕事を回せるようになり、

「仕事は楽でそれなりにアウトプットをだしているが全く成長はしない」
って一番危険な状態にはまりこむ。しかも給料は毎年あがる。

大企業はこの状態に陥りやすく私もこの状態で多くの時間を無駄にした。
同じ部署にいる場合は自分でどんどん新しい仕事に移る努力が必要

でも、そんなに部署も変わらないし、仕事は課長が決めるし、
社内公募もなかなかないよ っていうのが大多数の現実。

そこで、手っ取り早く「器用貧乏」になるための方法を紹介する

ツールや評価治具を積極的に作ろう！

利点はこんなにある

- ・会社の時間でこっそり自分の勉強ができる
- ・製品でないので失敗してもそれほどおこられない。
- ・自分が作りたいものを自分でつくるので
仕様作成・実装まで全て自分が好きなようにできる。
実装方法(フレームワーク、開発言語等)は自分で好きなものを選ぶ
つまらない社内調整や説明資料は最小限ですむ

FPGA、RasPiなどを使えば安価にいろいろなことが
簡単にできる時代です

チェックリストを作る作業より100倍おもしろいはず！

ツール職人の心得(1)

手段と目的をあえてひっくり返せ！

社会人になって上司から何度もいわれた言葉

「手段と目的を取り違えな。手段が目的になってはいけない」

→社会人としては正しいが技術習得の初期段階ではあえて無視する勇気が必要だと思う

例)

目的

センサ評価の時間を短くして、問題点を開発初期段階でフィードバックしたい。
効率的な開発ができプロジェクトにも貢献できる！

手段

- ・リアルタイム評価治具が必要
- ・FPGAを使えば実現可能だろう
- ・データ転送はPCIe or USB3.0を使おう



裏目的

- ・Linuxでデバイスドライバをかいてみたい。
- ・自分で設計したハードを汎用PCからアクセスしたい。
- ・マルチスレッドプログラムをかいてみたい。
- ・シェードプログラムもかいてみたい。
- ・ハード、ミドル、アプリ全部かけたら最高じゃん

やりすぎると社会人としてNG。会社とWin-Winの関係でいけるセンスが重要

ツール職人の心得(2)

車輪の再発明はあえて積極的に行え！
ただし、家に限る

車輪の再発明とは？

「広く受け入れられ確立されている技術や解決法を知らずに（または意図的に無視して）、同様のものを再び一から作ること」 by wiki

ソフトウェア業界では嫌われます。既存のライブラリや先行事例がある場合にそれを自分流でフルスクラッチで作ることはNGです。仕事では特に注意しましょう。

しかし、技術習得の過程では車輪の再発明はおそろしく勉強になります。ただし、これは家で自分の余暇を利用してやりましょう。会社で金をもらってやってツール職人として失格です。

ktuneの開発は完全に車輪の再発明です。

ツール職人の心得(3)

“Hello World” は全ての基本

- ・まずは**最小構成のコード**をビルドして動かすことが基本
- ・職業プログラマではないので無理に他人が書いたコードを流用する必要はない。
- ・ある程度まで自己流でOK。細かいことは気にするな。
コーディングルール等は職業プログラマになってからでOKのはず

自分でコードをかきコンパイルを自力で通すことで多くのことを学びます。
コードは下流の人間が書くものだ。上流の俺様には関係ないぜ！ って考えの人は
ツールは誰かに作らせて成果のみを上司に報告しましょう。考え方は人それぞれです。

FPGAの可能性について

- 私はFPGAが大好きです。
- 無限の可能性があると思います。
- ハード、ソフトの垣根はありません。
→どちらかできればいいという時代は終わりました。
- 開発環境も日々進化してます。
→高位合成, SoC, OpenXXなどぜんぜんついていけていません。
- 次の職場でもFPGAに関わると 생각합니다。

最後に世界のIntelがFPGAアプリケーションエンジニアなる職種を募集していたので紹介します。(ブログにまとめてくれていた人がいたので転記します)

職種

次世代のクラウドをFPGAでつくるためのFPGAアプリケーションエンジニア

仕事内容

Your primary responsibility will be optimizing different workloads using tightly coupled FPGA accelerators. Duties will include:

- Designing and developing RTL code using Verilog,
- Developing and debugging CPU Host SW that effectively offloads parts of workloads to FPGA,
- Developing OS and device drivers

仕事内容は、FPGAアクセラレータの開発。

- Verilog HDLでハードウェア実装したり、
- FPGAに仕事を割り振るホストCPU側のソフトを書いたり、
- それらのOSやデバドラを書いたりしてもらうよ

求められるスキル

- Experience in programming in C and/or C++ and assembly language,
- Experience in design, development and debugging of multithreaded programming code,
- Experience in hardware development using Verilog or System Verilog (or VHDL),
- Familiarity with FPGA design tools used for RTL development,
- knowledge of network protocols(TCP/IP),
- knowledge of storagetechnologies (RAID, file system),
- familiarity with revisioncontrol systems (GIT),
- Very good Linux OS understanding and experience in software development for Linux.
- Good English communication skills, both written and oral,
- Disciplined design approach, and ability to work smoothly with a team.

- C/C++とアセンブラ書けて、マルチスレッドの設計や開発は当然できて
- Verilog HDL/VHDLもすらすら書けて、FPGA開発ツールの経験もあって、
- TCP/IP熟知しつつ、RAIDやファイルシステムも理解し、
- 当然gitを日々使っていて、
- Linux OSの中身に精通し、
- 英語の会話も読み書きもすらすら、チームとのコミュニケーションもまかせとけっ

あると望ましいスキル

- Knowledge of CPU architecture, general GPU architecture,
- Prior experience working with heterogeneous (FPGA, CPU, GPGPU) hardware systems,
- Subject matter expertise in a particular class of algorithms, e.g. speech, cryptography, compression, image processing, etc,
- Activity in opensourcecommunity,
- Familiarity with computernetworks management (QoS, routing)
- OpenStack.

- 俺様CPUアーキ作れるぜ
- そーいやFPGA/CPU/GPUヘテロ開発の経験もあるし
- 音声認識・暗号化・圧縮・画像処理もやったっけ
- GitHubにもいろいろ上げてるし
- ネットワークのQoSやルーティングも書けるかな
- OpenStackね。知ってる。

こんなスーパーマンどこにいるんだろ？

やべー、少しでも近づきたい。

一日の大半、エクセルとパワーポイントを開いているってなにか違うよね。

40を超えたからって理由だけで技術から抜けるのはなんか違うくない？

管理の才能ないし、好きでもないし。

まだやりきってないよな。

まだ自分は成長できるはず



仕事は楽しいのが一番です