

Microsemi (旧 Actel) の SmartFusionFPGA を使うにあたってわけのわかんなかったところ、はまったところをメモしていきます (2011/10)

コンパレータを Fabric から使いたい

MSS Configurator から ACE を選択、ADD Service から Analog Comparator を追加すると MSS から外部に端子をエクスポートできます。

ACE FLAGS がややこしい

FPGA Fabric に引き出せる ACE FLAGS の元になるフラグ (画像参照) がややこしい。

- ・ FPGA_FLAGS を選択しないと MSS から外部に引き出すチェックボックスが表れないので注意。
- ・ 外部に引き出せるのは 32 bit のみ
- ・ 32bit の各々で THRESH_FLAGS を選択すると PPE_FLAGS0 | PPE_FLAGS1 | ... | PPE_SF_FLAGS の各ビットごとの OR になる。
- ・ PPE_FLAGSx or SF_FLAGS 自体 (0x40021450~60) はラッチされないが、引き出されるビットは立ち上がりエッジでラッチされるので注意。
- ・ ADC のところで「何 V 以上」とかいうフラグを設定して、ACE FLAGS を Fabric まで持ってきてもそれだけでは引き出した信号線は有効にならない。
 - ・ 該当するフラグの割り込み許可レジスタの該当ビットを ON しておかないとラッチされない。
 - ・ 0->1 の変化しかラッチされない
 - ・ ラッチされたビットはクリアしないと再検出できないが、ビットをクリアするためにはクリアレジスタへの書き込み操作が必要。
 - ・ Fabric からクリアするのが面倒な場合 (AHBlite or APB Master を作る必要がある)、Cortex-M3 側コードに自動生成される mss_ace ドライバから空の割り込みハンドラを登録すれば半自動でクリアできる。ただし、CPU の割り込み実行に依存するので、フラグのデアサートタイミング (= パルス幅) が非決定的になることに注意。

DEBUG の時 RAM が足りなくなる (SoftConsole Project)

SoftConsole を使うと、eNVM にコードを焼かずにコード、データ領域ともに eSRAM 上を取って SoftConsole からデバッグ実行できます。

eNVM には書き換え回数の制限がありますし、時間もかかるので、これはとても便利なのですが、コードも RAM に置くことになるので、あっという間に 64KB (A2F200) もある RAM を使い切っ

てしまいます。

SmartFusion に限ったことではないのですが、2 つほど注意することで RAM 使用量を減らせます。

1) コンパイラ、リンカオプションで不要な関数のリンクを避ける

- ・ MSS Configurator によって自動生成される firmware をライブラリ化せず、ソースツリーに直接取り込んでいる場合には特に有効です。
 - ・ コンパイラに `-ffunction-sections` オプションをつけ、関数ごとにセクションに分割する
 - ・ リンカに `--gc-sections` オプションをつけ、使っていないセクション (= 関数) を捨ててしまう

2) サンプルに含まれる、標準関数のサイズに注意

- ・ MSS Configurator が生成してくれるサンプルプロジェクトはとても便利なのですが、これに何か足して実験をしようとした場合に、使われている標準関数がメモリ不足に輪をかけていることがあります。
- ・ Map ファイルをよく見て、機能に比べてサイズの大きい標準関数は、自前の必要な機能に特化した関数に置換するとメモリの使用量を減らすことができます。
 - ・ たとえば、(s)printf 系関数がデバッグ出力に使われていることがあるため、そのままでは大量のメモリを消費します。
 - ・ デバッグ用の 16 進出力等だけであれば、さくっとその場で書くか、凝ったフォーマットであってもそのへんにある軽量 printf 系ルーチンで代替し、不要な形式のフォーマッタを削るようにすると、数キロバイトの節約になるようです (当社比 w)。

ModelSim の問題

・ MSS Configurator のクロック設定で外付け XTAL を接続する設定にすると、シミュレーション時 Fabric(FPGA) にクロックが供給されない

自動生成される `testbench.v` を見ると以下のようになっているので、

```
FOOBAR_top FOOBAR_top_0 (  
    // Inputs  
    ( 中略 )  
    .MAINXIN({1'b0}),
```

ModelSim には MAINXIN (クリスタル) にクロックが入っていないように見えるのが原因。

内蔵 RC オシレータの場合は内部 SmartFusion_MSS モジュールから供給されるので、一見動いているように見えるので気づかないことがある。

`testbench.v` を編集して適切にクロックを与えてやればよいが、このファイルは SmartDesign で再生成されるたびに上書きされてしまうので注意。

- ・ ModelSim 起動時に自動実行スクリプトの途中でエラーで止まる

いままで機嫌よく動いていたのに止まった場合は以下をチェック

- ・複数のインスタンスを起動しようとしていないか？（前回起動した ModelSim を落としていない等）
- ・シミュレーション対象にない信号が user.bfm に記述されていないか？ ロジックの名前を変えたのに、シミュレーション対象が古い名称のまま、等