

「FPGA ボードで学ぶ Verilog HDL」

補助テキスト Vol.2

2008/05/07

5E 小田原弘樹

第4章 実用回路の設計 1——180 秒タイマを作る

4.1 180 秒をカウントする

一言に「180 秒をカウントする」と言っても、回路の中ではちょっと複雑な処理をします。その流れは教科書 P59 の図 4-1 を見れば分かると思います。

教科書のリスト 4-1 の入出力ピンの定義で、**wire** と **reg** が使い分けられていることに気がついたと思います。wire と reg の違いですが、wire で宣言されたものを**ネット型変数**、reg で宣言されたものを**レジスタ変数**と言います。ネット型変数の信号への代入は assign 文でのみ可能で、レジスタ変数の信号への代入は always 文の中等でのみ可能です。

つまり、wire で宣言されたものに値を入れるには assign 文、reg で宣言されたものに値を入れるには always 文の中で、とすることです。また、**モジュール宣言されたものは自動的にネット型変数になる**ので、assign 文によって値を入れることが出来るのです。

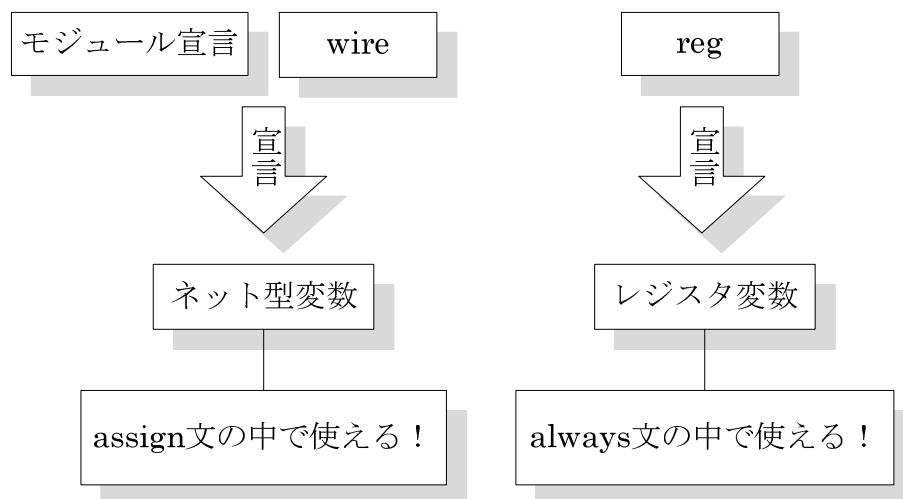


図 1 : wire と reg の違い

この授業で扱う FPGA ボードの 7 セグには、一度に違う数字を複数表示させることが出来ません。例えば 7 セグに“180”という数字を表示したい場合は、「1 桁目に 0 を出力→2 桁目に 8 を出力→3 桁目に 1 を出力」という処理を高速に連続に行うことによって、あたかも 7 セグに 180 と表示されているように見せるのです。この処理の記述は教科書 P63 に載っています。

ここの処理をもう少し詳しく解説すると、まず最初に“OE_DIGIT”という 2 ビットの変数を用意し

ます。この2ビットはクロック立ち上がりと共に1ずつカウントしていく“OEN_counter”の17番目と16番目の2ビットです。この2ビットのそれぞれの状態(00,01,10,11)によって、「X桁目にYを表示する」といった処理を行うのです。

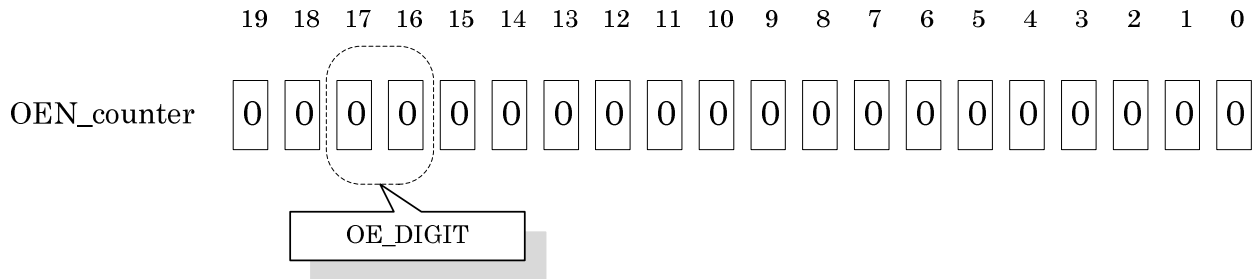


図2 : OEN_counter と OE_DIGIT の関係

教科書のリスト 4-1 最後の assign 文 4 つですが、これは今までに無い形の記述です。“LED1_O”の行を例にとって説明すると、“TIMEOUT”が“0”のとき、“Sec1_counter”の21ビット目が“1”の間点灯し、そうでない場合(“TIMEOUT”が“1”のとき)“Sec1_counter”の24ビット目が“1”の間点灯する、という意味です。“Sec1_counter”の21ビット目は、0.125秒ごとに“1”と“0”を繰り返すので、タイマーがカウントダウンを続けている間 LED1 は 0.125秒ごとに点滅するのです。

そして、“TIMEOUT”が“1”(カウントダウンが終了)になると4つのLEDは1秒ごとに点滅します。

```
assign LED1_O = ( TIMEOUT == 1'd0 ) ? Sec1_counter[21] : Sec1_counter[24]
```



図3 : リスト 4-1 末尾の assign 文

また、教科書 P66 の表 4-1 には単体 LED のピン配置が載っていないので注意しましょう。

- 【要点】**
- ①wire で宣言されたものは assign 文、reg で宣言されたものは always 文の中で値を入れる。
 - ②FPGA ボードの 7セグでは、一度に違う数字を複数表示させることができないので瞬間的に個々の数字を切り替えて表示させる。
 - ③教科書 P66 表 4-1 には単体 LED のピン配置が載っていない。P91 表 D-3 を参考に自分で定義すること。

第5章 実用回路の設計 2——デジタル時計を作る

5.1 デジタル時計の仕様を考える

教科書での“ブースト・モード”についての記述で、「4分が1秒で、4時間が1分で進むことになる」とありますがこれは恐らく誤りで、教科書のリスト 5-1 の記述では**その倍の早さで進んでしまいます**ので注意してください。

5.2 回路設計のポイント

教科書 P71 リスト 5-1 の「33MHz 時の 1/240 秒のタイミング」の所を見てください。ちょっと面倒くさい記述があります。「32'h0010C8E0 >> 4;」とありますが、これは 16 進の“10C8E80”という数を右に 4 ビットだけ論理シフト（右 4 ビットを切り捨て）する、という意味です。

実際にこれはどういう値なのか調べてみたいと思います。

10C8E80

↓ 2 進に直す

100001100100011100000

↓ 右に 4 ビットだけ論理シフト

10000110010001110

↓ 10 進に直す

68750

↓ クロック周波数 (33,000,000) で割る

0.00208333...

つまり、リスト 5-1 の通りに記述すると、ブースト・モードでは 0.00208333 秒 (1/480 秒) を 1 秒として刻むこととなります。よって **8分が1秒、8時間が1分の速さで進む**こととなります。

ブースト・モードの速さを変えてみましょう。例えば 10 分が 1 秒、という速さで進ませたい場合 0.00166667 秒を 1 秒として刻むこととなります。これに 33,000,000 を掛けた 55000 という値を “param_VMAXsec” に代入すれば良いので、以下のような記述となります。

```
parameter [31:0] param_VMAXsec = 32'd55000;
```

16 進だと分かりづらいので 10 進にしました。

教科書 P75 リスト 5-1 の一番上にある assign 文について、TIM_4 を例にとって説明します。これは “SW_DISPSEC_I” が “0” のとき、0 を 3 ビットと Count_10hour を足した 4 ビットの値を TIM_4 に代入するということです。例えば “Count_10hour” が “1” であれば、“0001” という値が代入されることとなります。

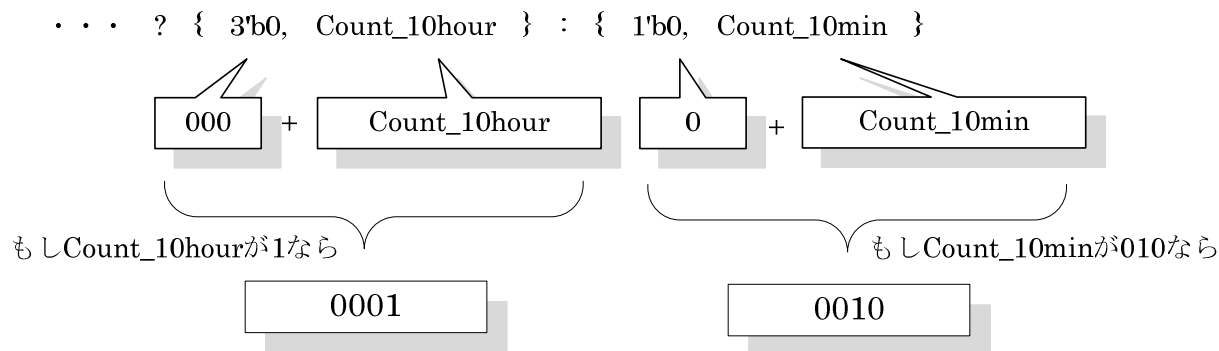


図 4 : リスト 5-1 “7 セグメント LED 表示部” の assign 文

また、教科書 P70 の表 5-1 には単体 LED のピン配置が載っていないので注意しましょう。

- 【要点】**
- ①教科書 P69 の“ブースト・モード”についての「4 分が 1 秒で、4 時間が 1 分で」という記述は誤り。リスト 5-1 の記述なら、「8 分が 1 秒、8 時間が 1 分」で進むことになる。
 - ②「32'h0010C8E0 >> 4」というような記述は、16 進の“10C8E0”という数を右に 4 ビットだけ論理シフト（右 4 ビットを切り捨て）するという意味。
 - ③教科書 P70 表 5-1 には単体 LED のピン配置が載っていない。P91 表 D-3 を参考に自分で定義すること。

付録 チャタリング除去回路

前回のテキストで予告した通り、チャタリングを除去する回路の記述法を紹介します。

教科書 P46 リスト 3-5 のプログラムを例にとってチャタリング除去回路を組み込んで見たいと思います。このプログラムはスイッチを押した回数を 7 セグに表示するというものですが、チャタリングが生じてしまうと一気に何個か数字が飛んでしまうので、チャタリングの確認にはちょうど良いプログラムです。リスト EX-1 にチャタリングを防止したスイッチカウンタのプログラムを載せたので組んでみてください。

リスト EX-1 チャタリング除去回路を組み込んだスイッチカウンタ

```

module test009(
    reset,
    switch,
    seg7_oe_n,
    seg7_n,
    clk
);

input reset;
input switch;
input clk;

output [4:1] seg7_oe_n;
output [8:1] seg7_n;

reg [3:0] counter4;
reg [7:0] reg_seg7;

```

```

parameter count_clk_max = 'd1500000; // =0.0455 秒です(1,500,000/33,000,000)

reg [20:0] count_clk; // 21 ビットなので 2,097,152 までカウントできます

always @(posedge clk) // クロック立ち上がりするとき
begin
  if(reset == 1'b1) // リセットスイッチが押されたら
  begin
    count_clk <= 0; // クロックのカウンターと
    counter4 <= 0; // スイッチのカウンターをリセット
  end else
  begin
    if(switch == 1'b1) // カウントスイッチが押されたら
    begin
      if(count_clk <= count_clk_max) // クロックのカウントを開始
      begin
        count_clk <= count_clk + 1;

        if(count_clk == count_clk_max) // スイッチが一定時間(0.0455 秒)押されていたら
        begin
          counter4 <= counter4 + 1; // やっとこさスイッチをカウントする
        end
      end
    end else
    begin
      count_clk <= 0; // カウントスイッチが押されていないときは
    end // クロックのカウントは 0
  end
end

assign seg7_oe_n = 4'b1110;

always @*
begin
  case(counter4)
    4'b0000 : reg_seg7<=8'b00111111;
    4'b0001 : reg_seg7<=8'b00000110;
    4'b0010 : reg_seg7<=8'b01011011;
    4'b0011 : reg_seg7<=8'b01001111;
    4'b0100 : reg_seg7<=8'b01100110;
    4'b0101 : reg_seg7<=8'b01101101;
    4'b0110 : reg_seg7<=8'b01111101;
    4'b0111 : reg_seg7<=8'b00100111;
    4'b1000 : reg_seg7<=8'b01111111;
    4'b1001 : reg_seg7<=8'b01101111;
    4'b1010 : reg_seg7<=8'b01110111;
    4'b1011 : reg_seg7<=8'b01111100;
    4'b1100 : reg_seg7<=8'b01011000;
    4'b1101 : reg_seg7<=8'b01011110;
    4'b1110 : reg_seg7<=8'b01111001;
    4'b1111 : reg_seg7<=8'b01110001;
  endcase
end

assign seg7_n = ~reg_seg7;

endmodule

```

制約ファイルの記述は表 EX-1 の通りです。

表 EX-1

NET "seg7_oe_n[4]"	LOC="P2";
NET "seg7_oe_n[3]"	LOC="P3";
NET "seg7_oe_n[2]"	LOC="P85";
NET "seg7_oe_n[1]"	LOC="P86";
NET "seg7_n[8]"	LOC="P95";
NET "seg7_n[7]"	LOC="P91";
NET "seg7_n[6]"	LOC="P90";
NET "seg7_n[5]"	LOC="P70";
NET "seg7_n[4]"	LOC="P71";
NET "seg7_n[3]"	LOC="P98";
NET "seg7_n[2]"	LOC="P92";
NET "seg7_n[1]"	LOC="P94";
NET "switch"	LOC="P27";
NET "reset"	LOC="P30";
NET "clk"	LOC="P63";

このプログラムが元のプログラムと何が違うかと言うと、「スイッチが一定時間押されていないとカウントしない」というところです。リスト EX-1 のプログラムでは、スイッチが 0.0455 秒以上押されていないとカウントしません。

チャタリングは微小な振動です。これによってスイッチが押されている時間は、バネの性質によって異なりますがせいぜい数 μ ~ 数 m 秒といったところなので、このプログラムではそのような短時間のスイッチングは相手にしません。一般的なスイッチ操作をしていれば問題ありませんが、もし高橋名人並みの連打が出来る人がいればチャタリングと見なされてカウントされないかも知れません。

リスト EX-1 のプログラムがちゃんと動作してチャタリングの除去を確認出来たら、他のプログラムにもチャタリング除去プログラムを組み込んでみましょう。

以上で「FPGA ボードで学ぶ VerilogHDL」の解説を終わります。

編集後記

拙い文章で本当にごめんなさいでした。これでも頑張った方なんです。一応。

しかしまあテキストを作るって難しいですね。何が難しいって頭の中にあることを文章や図にすることがですよ。きつこういふのは慣れなんでしょうね。普段文章を書くことなんてほとんど無いからもう悩みまくりです。でもいい経験になりました。Verilog についての知識も少し身に付いたし。

まあ、こんなテキストでもちょっとは皆さんにとってプラスになったらなあと思います。

あ、もし「こんなテキストなんかよりも、オレ様がもっと分かりやすいテキスト作ってやるぜうへへ」という人がいたら是非この仕事受け継いで下さい。伊藤先生か山崎先生あたりに言えばやらせてもらえると思います。

こんなとこですかね。工学研究の時間ちょっとお邪魔するかも知れませんが、その時はあまりイヂメないで下さい。楽しくやりましょう。楽しく。