

FPGA の回路設計

1 目的

FPGA (Field Programmable Gate Array) は「現場でプログラム可能な論理ゲート配列」という意味であり、ユーザーが希望する論理機能を短時間で実現でき、しかも何度も書き換えが可能な計算デバイスである。近年、安価で高速に動作する FPGA が開発されたこと、プログラミングが容易になったことなどから注目され、様々な用途に使用されている。本実験では、FPGA を始めとする書き換え可能な計算デバイスを用いる意義を理解し、実際に使用してみることを目的とする。

2 解説

2.1 歴史

既に学んでいるように、初期においては、トランジスタや抵抗を用いて論理ゲートを構成し、これらをまとめて論理回路を作っていた。その後、IC (Integrated Circuit: 集積回路) の開発により、論理ゲートは IC 内に収まるようになった。さらに、自由な回路を設計するために PLA (Programmable Logic Array) や、PAL (Programmable Array Logic) が開発され任意の機能を持つ論理回路を IC 内に構成できるようになった。この考えを大規模にし、内部回路を何度も修正可能にしたものが FPGA である。

2.2 FPGA の位置づけ

FPGA は、特定処理ごとの専用ハードウェア ASIC (Application Specific Integrated Circuit) と、Pentium シリーズのような汎用プロセッサの中間的な位置づけであるといえる。ASIC は高速に動作可能で消費電力が少ないという長所を持つが、用途毎に設計する必要があり、設計・開発にコストと時間がかかるという欠点がある。汎用プロセッサはソフトウェアの変更だけ様々な用途に使用できるが、動作が比較的低速で、消費電力も大きい。それに対して、FPGA は両者の長所を併せ持ち、ASIC には及ばないものの、低消費電力かつ高速な処理が可能である。また、ソフトウェアによって動作を容易に変更でき、最近は C 言語のプログラムを FPGA に実装できるようになっている。

2.3 FPGA の基本的な動作原理

FPGA の特徴である Logic Cell (LC) について説明する [1]。基本的な LC は図 1 のように Look-Up Table (LUT) と、フリップフロップ (Flip Flop) から構成される。この例では、LC の入力は $I_1 \sim I_4$ 、出力は Q である。まず LUT であるが、 $I_1 \sim I_4$ を入力した時の LUT の出力 O はプログラムによって自由に設定可能で、任意の関数を実現できる。表 1 は 4 入力 OR を実現した時の LUT である。LUT の出力 O はフリップフロップに入力され、値を保持する。そして、同期信号が high (1) になった時に保持した値 Q を出力する。

さらに LC を様々なに結線して組み合わせることで、全体として任意の回路を実現する。

表 1: Look-Up Table の例 (4 入力 OR の場合)

I_1	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
I_2	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1
I_3	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1
I_4	0	1	1	1	0	1	1	1	0	1	1	1	0	1	1	1
O	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

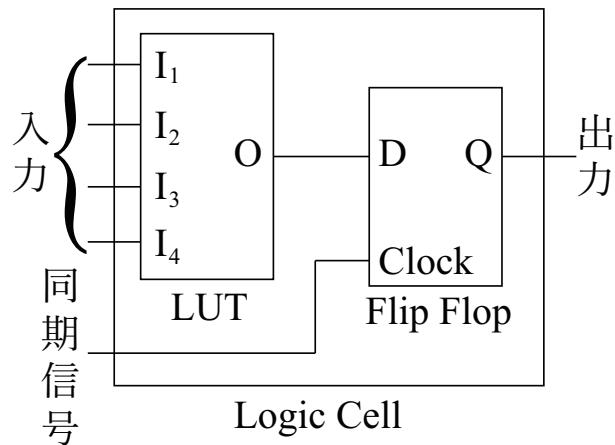


図 1: Logic Cell の基本的な構成 [1]

3 実験の解説

3.1 実験機材

本実験で用いる機材は、FPGA チップとして、Intel 社製 MAX 10 10M50DA を用いる。このチップは Terasic 社製 FPGA ボード DE10-Lite に載っている。De10-Lite は PC に接続されている。

3.2 FPGA チップの概要

FPGA チップは図 2 に示すように、Configurable Logic Blocks (CLBs), Programmable Input/Output Blocks (IOBs), Delay-Locked Loops (DLLs) および Block RAM から成る。外側の小さい四角は信号および電源用の取り出し部分で、この部分とパッケージピンがリード線で結線されている。また、上下および左右方向にバスが走っており、CLB 間および CLB と IOB 間の結線も自由に設定できる。

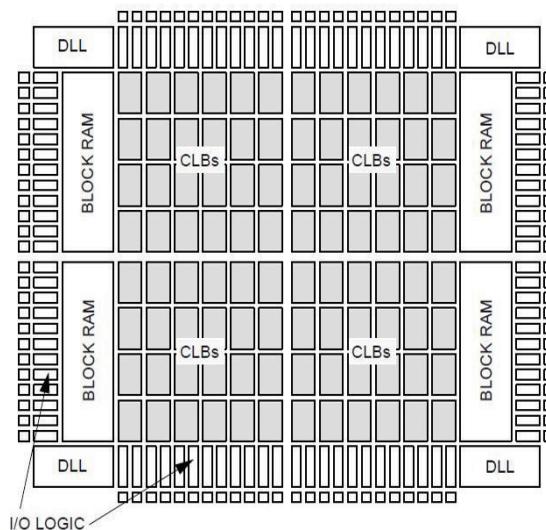


図 2: FPGA の IC 内部配置図¹(図は Xilinx 社製 XC2S15 のもの)

¹<http://www.seekic.com/uploadfile/ic-mfg/2013113214829464.jpg> より

3.2.1 Configurable Logic Blocks (CLBs)

XC2S15 では、2.3 で述べた Logic Cell (LC) を 4 つ持つ CLB が基本ユニットであり、LC を単独で用いるより大きな値を扱うことや複雑な演算をすることができる。1 つの CLB は図 3 に示すスライス 2 つから成り、1 つのスライスは LC を 2 つ持つ。図 3 の上下に 1 つずつある LC は、図 1 のものとは異なり、LUT とフリップフロップの間に Carry and Control Logic を持つ。これにより、加算器として使用することや、上下 2 つのユニット間で信号選択ができる。

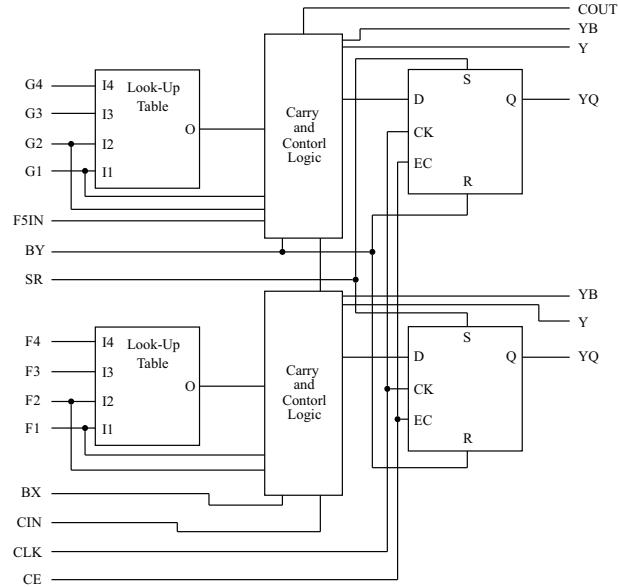


図 3: CLB のスライスの構成

3.2.2 Programmable Input/Output Blocks (IOBs)

IOB はパッケージピンと内部の回路の接合部である。プログラマブルであることにより、各種の I/O 規格に対応できる。

3.2.3 Delay-Locked Loops (DLLs)

DLL は同期のために、水晶発振器からのチップに入力されたクロックを倍増(周波数を n 倍に変換すること)、分周(周波数を $1/n$ に変換すること)し、チップレベルとボードレベルのクロックを制御する装置である。

3.2.4 Block RAM

FPGA を実現するためには、LUT の内容を記憶しておく必要がある。LUT は CLB に内蔵されているので、記憶領域を CLB 内部に置くこともできるが、これでは空間効率が良くない。そこで XC2S15 では LUT を実現するためのメモリを集積している。

3.3 FPGA の回路設計とコンフィグレーション

通常、回路の設計は以下の 6 段階で行う。

1. 設計 : Intel 社から配布されている Quartus Prime を用いて回路を設計し、HDL (Hardware Description Language) で記述される回路を得る。直接 HDL で記述して回路を設計することもできる。
2. シミュレーション : 実機で動作させる前に、シミュレータを使って動作確認をする。
3. ピンアサイン設定 : 回路図あるいは HDL 記述における入出力と FPGA ボード上のピンとを対応付ける。各ノート PC に備え付けのファイル「DE10-Lite.pin.qsf」(デスクトップ上にある)を使うこと。
4. コンパイル : 設計した回路に不備がないかチェックする。コンパイルエラーが生じた場合、修正する。
5. コンフィグレーション : コンパイルによって得られた sof 形式のファイルの内容を FPGA に書き込む。
6. 実行 : 書き込み終了後、回路は自動的に動作を開始する。

4 実験

4.1 Quartus Prime の練習

練習 1 デスクトップ上にあるファイル (参考資料) 「reference.pdf」を参考に、7 セグメント LED デコーダを、HDL を記述することによって作成し、実機にて動作確認せよ。作成回路は、4 ビットのスイッチ入力に対し、7 ビットの 7 セグメント出力を定義しており、スイッチ入力 (SW[0]~SW[3]) の値に応じて 16 通りに分岐し、HEX0 に表示パターンを代入しています。動作確認後、井上のチェックを受けること。

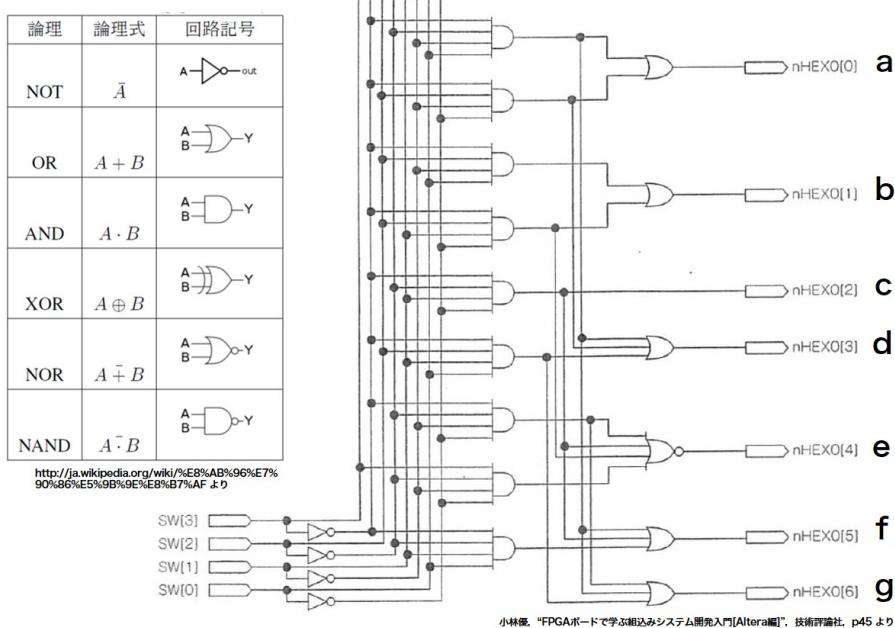


図 4: 7 セグメント・デコーダ回路図

【注意事項】

1. スイッチは、下側のときは 0 で、上側のときは 1 を表す。
2. LED は、0 のときは点灯し、1 のときは消灯する。直感とは逆なので注意。

練習 2 参考資料「reference.pdf」を参考に、10秒カウンタを、HDLを記述することによって作成し、実機にて動作確認せよ。動作確認後、井上のチェックをうけること。

練習 3 以下の2つのHDLを参考に、アップダウンカウンタを、HDLを記述することによって作成し、実機にて動作確認せよ。以下のHDLファイルは、[4]記載のものをDE10-Lite版へ書き換えたものである。動作確認後、井上のチェックを受けること。

```

module UPDOWN(
    input    CLK, RST,
    input    [1:0] KEY,
    output   reg [6:0]  HEX0
);

wire      reset, down, up;
BTN_IN BTN_IN(CLK, RST, KEY, {reset, down, up});

reg [3:0] udcnt;
always @(posedge CLK, posedge RST) begin
    if(RST)
        udcnt <= 4'h0;
    else if(up)
        udcnt <= udcnt + 4'h1;
    else if(down)
        udcnt <= udcnt - 4'h1;
end

always @* begin
    case (udcnt)
        4'h0:   HEX0 = 7'b1000000;
        4'h1:   HEX0 = 7'b1111001;
        4'h2:   HEX0 = 7'b0100100;
        4'h3:   HEX0 = 7'b0110000;
        4'h4:   HEX0 = 7'b0011001;
        4'h5:   HEX0 = 7'b0010010;
        4'h6:   HEX0 = 7'b0000010;
        4'h7:   HEX0 = 7'b1011000;
        4'h8:   HEX0 = 7'b0000000;
        4'h9:   HEX0 = 7'b0001000;
        4'ha:   HEX0 = 7'b0000100;
        4'hb:   HEX0 = 7'b0000011;
        4'hc:   HEX0 = 7'b1000110;
        4'hd:   HEX0 = 7'b0100001;
        4'he:   HEX0 = 7'b0000110;
        4'hf:   HEX0 = 7'b00001110;
        default: HEX0 = 7'bxxxxxxxx;
    endcase
end
endmodule

```

(a) updown.v

```

module BTN_IN(
    input    CLK, RST,
    input    [2:0] nBIN,
    output   reg[2:0] BOUT
);

reg [20:0] cnt;
wire en40hz = (cnt==1250000-1);

always @(posedge CLK, posedge RST) begin
    if(RST)
        cnt <= 21'b0;
    else if(en40hz)
        cnt <= 21'b0;
    else
        cnt <= cnt + 21'b1;
end

reg [2:0] ff1, ff2;

always @(posedge CLK, posedge RST) begin
    if(RST) begin
        ff2 <= 3'b0;
        ff1 <= 3'b0;
    end
    else if(en40hz) begin
        ff2 <= ff1;
        ff1 <= nBIN;
    end
end

wire [2:0] temp = ~ff1 & ff2 & {3{en40hz}};

always @(posedge CLK, posedge RST) begin
    if (RST)
        BOUT <= 3'b0;
    else
        BOUT <= temp;
end
endmodule

```

(b) btn_in.v

図5: アップダウンカウンタのHDLファイル

4.2 任意回路の作成

興味のある動作をする回路を設計し、FPGA内で構成し、その動作を確認せよ。動作確認後、井上のチェックを受けること。「reference.pdf」の第3章を参考にして良いが、3章の回路そのものを提出しても評価の対象としない。レポートには、何を作ろうとして、どうなったか(結果)を書くこと。うまくいかなかったからと言って減点しないが、それによってレポートがきちんと書けていない場合は減点する。

【過去にあった例】

- 10進カウンター
- ストップウォッチ
- 文字がスクロールする広告みたいなやつ
- 「2進数→16進数」変換器

5 課題

以下をレポートにまとめ、提出せよ。

なお、作成した回路図や HDL のファイルは授業支援システムを通じて提出すること。

1. FPGA がどのような用途に使用されているのか調べよ。
2. 練習 1 の 7 セグメントデコーダにおいて、「reference.pdf」の表 2-2 の真理値表(本資料の図 6)から、図 4 の回路図の導出を、(カルノー図等(この他でも良い)を用いて) レポートに記すこと。
3. 4.2 で作成した回路の想定回路図、動作確認の手順及び結果を報告せよ。動作確認は授業中に行います。
4. 各項目において、苦労した点や工夫した点などを報告せよ。

6 その他

- ピンアサイン設定のためのファイル「DE10-Lite.pin.qsf」は各ノート PC のデスクトップに置いてある。
- デスクトップ上に「学籍番号」というフォルダを作成し、そのフォルダ内に実験用データを全て保存すること。また、課題毎にサブフォルダを作成し、データを保存すること。他のフォルダにはデータを保存しない。
- 参考資料「reference.pdf」は、小林優「改訂 2 版 FPGA ボードで学ぶ組込みシステム開発入門 [Intel FPGA 編]」(技術評論社) [5] の 1 章から 3 章の抜粋である。
- 参考のため、本資料の末尾に表 2：論理回路の回路記号と Verilog HDL の文法例²を示す。
- 使用 PC は、本実験用に環境を整えている。環境保持のため、**ネットには接続しないこと**。ネットを用いて検索したい場合は、各自のノート PC もしくはスマートフォン等で検索すること。

7 成績の付け方

- レポートの提出期限、提出場所は下記の通りである。ただし、B4 棟は午後 7 時ごろに施錠されるので、注意すること。
 - 提出期限：実験最終週の翌週の火曜日まで
 - 授業支援システム (Moodle) の情報工学実験 1 の FPGA の項目のところから提出すること。
- レポートは、「4. 実験」と「5. 課題」についてまとめること。本資料に記載してあることを重複して書く必要はない。
- 結果を羅列しただけでなく、考察がきちんと書けているレポートを高く評価する。
- レポートは手書きでもパソコンで作成しても良い。
- レポートを作成する上で書籍や Web ページを参考にした場合は、参考したものを記載すること。この記載がなく、ほぼ同一の記述を発見した場合は大きく減点する。
- レポートが他人のものに著しくしているなど、他人のレポートを写したと判断した場合は、写した側も書かれた側も大きく減点する。

²資料内の\$display は、実機を伴わず FPGA の動作確認するためにシミュレータで用いるもので、実機では動作しないことに注意すること。

▼表 2-2 7セグメント LED デコーダの真理値表

入力	表示	出力 (gfedcba)	入力	表示	出力 (gfedcba)
0000	0	1000000	1000	8	0000000
0001	1	1111001	1001	9	0010000
0010	2	0100100	1010	A	0001000
0011	3	0110000	1011	b	0000011
0100	4	0011001	1100	C	1000110
0101	5	0010010	1101	d	0100001
0110	6	0000010	1110	E	0000110
0111	7	1011000	1111	F	0001110

出力は0で点灯、1で消灯

図 6: reference.pdf 内の表 2-2 記載の真理値表

表 2: 論理ゲート ([3] より転載)

論理	論理式	回路記号
NOT	\bar{A}	
OR	$A + B$	
AND	$A \cdot B$	
XOR	$A \oplus B$	
NOR	$A + \bar{B}$	
NAND	$A \cdot \bar{B}$	

参考文献

- [1] 今さら聞けない FPGA 入門
<http://monoist.atmarkit.co.jp/mn/articles/0609/20/news118.html>
- [2] Verilog HDL 入門
<http://research.kek.jp/people/uchida/educations/verilogHDL/>
- [3] フリー百科事典『ウィキペディア (Wikipedia)』, 「論理回路」の項
<http://ja.wikipedia.org/wiki/%E8%AB%96%E7%90%86%E5%9B%9E%E8%B7%AF>
- [4] 小林優, FPGA ボードで学ぶ組込みシステム開発入門 [Altera 編], 技術評論社
- [5] 小林優, FPGA ボードで学ぶ組込みシステム開発入門 [Intel FPGA 編], 技術評論社