

雑誌記事の活用

2時間でCDプレーヤを作成

田中良平

本稿執筆中に、筆者の手元に届いた書籍に、パソコン用のCD-ROMドライブを活用する音楽CDプレーヤの製作記事がありました¹⁾。記事には、ATA インターフェースを使ってCDを演奏

するために必要なATAPIの説明と、CDプレーヤの設計思想が記述されており、興味を持ちました。しかしATA/ATAPIの原理を理解して、自分で同じものを設計するより先に、「動かしてみたい」といった興味のほうが先に立ちました。この書籍には、記事のほかに設計データが収録されたCD-ROMが付属していたので、VHDLソースを使用してみました。

ちょうど筆者の手元に、ATAインターフェース回路(FPGAの外付け回路)があらかじめ用意されているFPGA開発ボードがありました。搭載されているFPGAは、Xilinx社のVirtex-IIです。しかし書籍の記事ではAltera社のFLEX10KE向けに記述されていました。このため、若干のカスタマイズが必要になります。

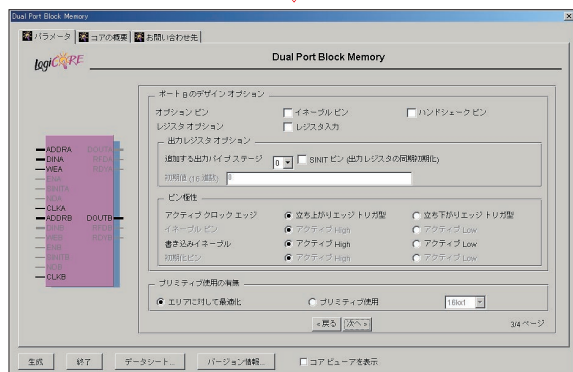
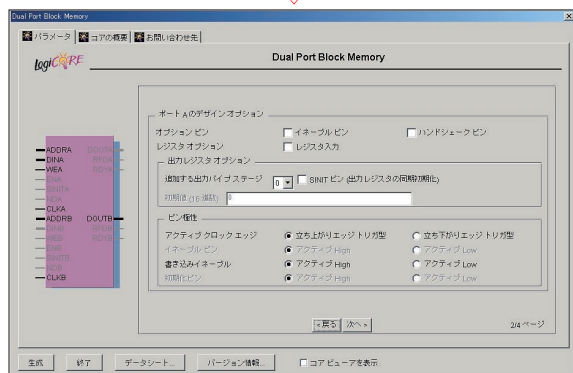
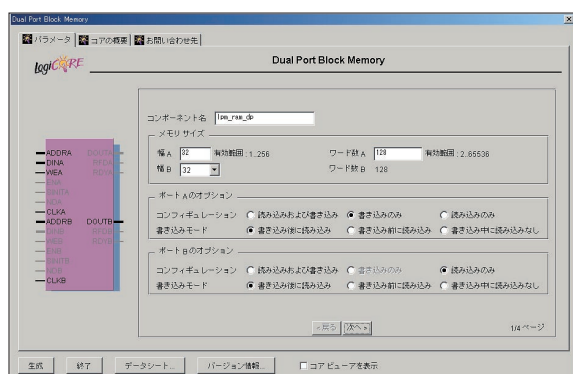
カスタマイズといっても、全体的にVHDLで記述されているので、それほど大きな変更は必要ありません。デバイス固有の機能、具体的にはFPGA内蔵のメモリを使用するための記述を、Altera社向けからXilinx社向けに変更するだけです。

Xilinx社向けのコード生成は、設計ツールISEが持つCORE Generatorの機能を使用して行います(図A)。オリジナルのソース・コードからアドレス幅7ビット、データ幅32ビットのデュアルポート・メモリであること、ポートAは書き込み専用、ポートBは読み込み専用であることがわかるので、同様の機能になるようにパラメータを設定します。メモリ・コアの名称は便宜上、オリジナルと同じにしました。

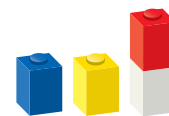
リストAのようにVHDLではcomponent宣言部分を入れ替えて、続いて、リストBのようにport mapを入れ替えることでカスタマイズは完了します。

オリジナルの回路は、FPGAの周辺回路として接続されている押しボタン・スイッチを使用して演奏モード(再生、停止、早送りなど)を切り替えるようになっていました。すなわち、ボタンが押されたことを示す信号さえ入力すれば、音楽CDプレーヤとしての複雑な制御はすべて実行してくれます。ですから、今回はこのコアのボタンの代わりに組み込み用のCPUコアを置いて、そのI/Oポートから制御するようにしました。図Bのようにわずか6本の線でCDプレーヤを制御できてしまいます。

せっかくCPUを使っているので、リモートで制御できるようにCPUにシリアル通信用のIPコアも追加しました。CPUのプログラムとしては、受信したコマンドを解釈してI/Oポートを上下させるといった単純なものです。

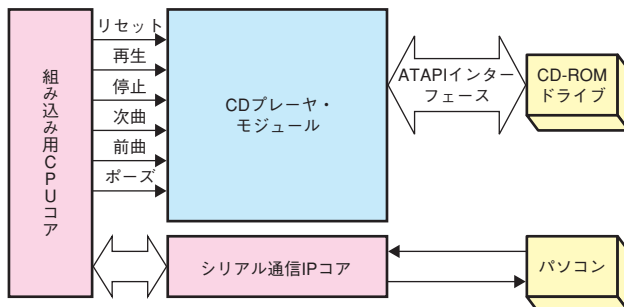


〔図A〕CORE Generatorでメモリを作成



〔リストA〕 component 文の置き換え

オリジナル (Altera 社 PLD 向け)		Xilinx 社 FPGA 向け
<pre> component lpm_ram_dp generic(LPM_WIDTH : positive ; LPM_WIDTHAD : positive ; LPM_NUMWORDS : natural := 0 ; LPM_TYPE : string := "LPM_RAM_DP" ; LPM_INDATA : string := "REGISTERED" ; LPM_OUTDATA : string := "REGISTERED" ; LPM_RDADDRESS_CONTROL : string := "REGISTERED" ; LPM_WRADDRESS_CONTROL : string := "REGISTERED" ; LPM_FILE : string := "UNUSED" ; LPM_HINT : string := "UNUSED") ; port(rdaddress, wraddress : in std_logic_vector(LPM_WIDTHAD-1 downto 0) ; rdclock, wrclock : in std_logic := '0' ; rden, rdclken, wrclken : in std_logic := '1' ; wren : in std_logic ; data : in std_logic_vector(LPM_WIDTH-1 downto 0) ; q : out std_logic_vector(LPM_WIDTH-1 downto 0)) ; end component ; </pre>	→	<pre> component lpm_ram_dp port (addra : IN std_logic_VECTOR(6 downto 0); addrb : IN std_logic_VECTOR(6 downto 0); clka : IN std_logic; clkb : IN std_logic; dina : IN std_logic_VECTOR(31 downto 0); doutb : OUT std_logic_VECTOR(31 downto 0); wea : IN std_logic); end component; </pre>



〔図B〕全体の接続ブロック図

カスタマイズも含めて、要した時間は2時間程度でした。すべて完成されたモジュールをつなげただけなので、すぐに音楽CDを再生することができました。

おそらく同様の機能を自分で作ろうとすると、この何十倍、何百倍もの時間が必要になることでしょう。ATAPIについての知識がなくても、このように、すでに実績のある機能を使うことで目的を達成できれば、設計期間を大幅に短縮できます。

一度、動くことが確認できれば、ATA/ATAPIについて勉強する意欲もわいてきます。さらに、この回路を参考にしながら、自分の要求に合わせた拡張も行いやすくなると思います。

参考文献

- 1) 河野崇, 「CD プレーヤの製作」, 『FPGA/PLD 設計スタートアップ』, pp.128-144, CQ 出版.

〔リストB〕 port map 文の置き換え

オリジナル (Altera 社 PLD 向け)
<pre> TOCMEM: lpm_ram_dp generic map(LPM_WIDTH => 32, LPM_WIDTHAD => 7, LPM_WRADDRESS_CONTROL => "REGISTERED", LPM_RDADDRESS_CONTROL => "UNREGISTERED") port map(rdaddress => RAdrs, wraddress => WAdrs, rdclock => CLK, wrclock => CLK, rden => Active, wren => WEn, data => WData, q => RData) ; </pre>



Xilinx 社 FPGA 向け
<pre> TOCMEM : lpm_ram_dp port map (addra => WAdrs, addrb => RAdrs, clka => CLK, clkb => CLK, dina => WData, doutb => RData, wea => WEn); </pre>

たなか・りょうへい

DAIHEN Advanced Component 社