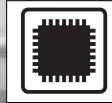


# 基礎から学ぶ Verilog HDL & FPGA 設計

## 第8回

## 分散 RAM とブロック RAM

中野浩嗣, 伊藤靖朗



デバイスの記事



ビギナーズ

今回は、CPUの重要な構成部品であるメモリとバスを設計する。最終的に、CPUの骨格である機械語命令フェッチ回路を設計し、シミュレーションとFPGAボードを用いた動作確認を行う。  
(筆者)

### ● FPGAの基本構造

Xilinx社のFPGA Spartan-3EやSpartan-3Aは、次の五つのプログラム可能な構成要素を持ちます<sup>(1),(2)</sup>。

#### コンフィギャラブル・ロジック・ブロック(CLB)

コンフィギャラブル・ロジック・ブロック(CLB)は、組み合わせ回路や順序回路をインプリメントするためのエレメントです。CLBは四つのスライス(Slice)から構成されます。各スライスは、組み合わせ回路、フリップフロップ、またはシフト・レジスタのいずれかをインプリメントすることができ、16ビットの大きさ(4入力1出力)のルックアップ・テーブル(LUT)を二つ含みます。

#### 入出力ブロック(IOB)

入出力ブロック(IOB)は、FPGAの入出力ピンと内部の回路とのデータ・フローを制御します。

#### ブロックRAM(Block RAM)

ブロックRAM(Block RAM)は、大きさ18Kビットのデュアルポート・メモリです。アドレス・バスやデータ・バスの幅や初期値を自由に設定できます。

#### 乗算器

乗算器は二つの18ビットの整数を乗算できます。

#### デジタル・クロック・マネージャ(DCM)

デジタル・クロック・マネージャ(DCM)は、クロッ

ク信号の分配や遅延制御などを行います。

Spartan-3Eスタータキットに搭載されているFPGA(XC3S500E-4FG320C)は、1164個のCLB、20個のブロックRAM、および、20個の乗算器を持ちます。20個のブロックRAMをすべて用いると、最大18Kビット×20=360Kビットの記憶容量を持つことになります。

メモリのインプリメント方法には、コンフィギャラブル・ロジック・ブロックのルックアップ・テーブルを用いる方法もあります。このインプリメント方法によるメモリは分散RAMと呼ばれます。XC3S500E-4FG320Cの場合、分散RAMは理論上、16ビット×4×1164=74496ビットの記憶容量を持ちます。しかし、コンフィギャラブル・ロジック・ブロックは組み合わせ回路や順序回路のインプリメントにも用いられるので、実際に利用できる容量は少なくなります。

後で述べるように、分散RAM(Distributed RAM)は非同期読み出しが可能です。ブロックRAMは同期読み出ししかできません。このため、ブロックRAMの方が扱いが難しくなります。従って、大容量の場合はブロックRAMを用い、小さい場合は分散RAMを用いるといった使い分けがよいでしょう。本連載で設計するCPUで、機械語プログラムやデータを格納するメモリはブロックRAMを用います。

### ● 分散RAMの設計

ブロックRAMを設計する前に分散RAMを設計してみましょう。リスト1のVerilog HDL記述を、ISE WebPACK

#### Keyword

メモリ, バス, 機械語命令フェッチ回路, 分散RAM, ブロックRAM, レジスタ型変数

で論理合成を行うと、分散RAMに変換されます。

入力ポートは、clk, load, addr, dの四つであり、出力ポートはqだけです。

2行目のパラメータで、DWIDTHとAWIDTHのデフォルト値を両方とも8にしています。これらはそれぞれ、データとアドレスのビット幅を指定しています。また、WORDSはデータの個数であり、アドレスが8ビットなので $2^8 = 256$ としています。

入力ポートaddrはアドレスを指定するのに用い、そのビット数はAWIDTH、つまり8です。

入力ポートdは書き込むデータを入力するのに用います。そのビット数はDWIDTH、つまり8です。

出力ポートqは読み出したデータを出力するのに用いられ、そのビット数はDWIDTHです。

8行目で、レジスタ型変数memを宣言しています。memは、WORDS個の要素からなる1次元配列で、その各要素はDWIDTHビットです。つまり、mem[0], mem[1], ..., mem[WORDS-1]は、それぞれDWIDTHビットのレジスタ型変数とみなすことができます。

10行目のalways文で、clkの立ち上がりごとに11行目のif文を実行します。このif文は、loadが'1'のとき、mem[addr]にdの値を書き込みます。

また、13行目のassign文でメモリの読み出しを定めています。出力ポートqには、常にmem[addr]の値が継続的に出力されます。書き込みはクロックclkの立ち上がりに同期して行われ、読み出しはクロックに関係なく、addrで指定した番地の値が常に出力されるので、同期書き込み・非同期読み出しメモリと呼ばれます。このようなメモリは、ルックアップ・テーブルを用いた分散RAMにインプリメ

ントされます。

実際、dram.vをトップ・モジュールとした後、Processウィンドウの「Synthesis-XST」の階層の下にある「View Synthesis Report」をダブル・クリックして論理合成結果を見ると、

```
256x16-bit single-port distributed RAM      : 1
という記述が見つかるはずで
```

256個の16ビット・ワードを持つシングル・ポート分散RAMに変換されていることが分かります。また、論理合成結果にある、

```
Number of Slices:                85 out of 4656    1%
Number of 4 input LUTs:          168 out of 9312    1%
Number used as logic:             40
Number used as RAMs:              128
```

から、ルックアップ・テーブルのうち、128個がRAMとして用いられていることが分かります。一つのルックアップ・テーブルの容量は16ビットなので、その容量は16ビット×128=2048ビットです。従って、分散RAMの容量8ビット×256=2048ビットと一致しています。

## ● ブロックRAMの設計

本連載で設計するCPUで実際に用いるのは、ブロックRAMです。そのためには、同期書き込み、同期読み出しメモリを設計する必要があります。リスト2はそのVerilog HDL記述です。

入出力ポートは分散RAMと同じです。8行目でqをDWIDTHビットのレジスタ型変数として宣言しています。11行目に始まるalways文で、書き込みと読み出しの動作を定めています。クロックclkの立ち上がりごとに、13行目

リスト1 分散RAMのVerilog HDL記述(dram.v)

```
1 module dram(clk, load, addr, d, q);
2   parameter DWIDTH=8, AWIDTH=8, WORDS=256;
3
4   input clk, load;
5   input [AWIDTH-1:0] addr;
6   input [DWIDTH-1:0] d;
7   output [DWIDTH-1:0] q;
8   reg [DWIDTH-1:0] mem [WORDS-1:0];
9
10  always @(posedge clk)
11    if(load) mem[addr] <= d;
12
13  assign q = mem[addr];
14
15  endmodule
```

リスト2 ブロックRAMのVerilog HDL記述(ram.v)

```
1 module ram(clk, load, addr, d, q);
2   parameter DWIDTH=16, AWIDTH=12, WORDS=4096;
3
4   input clk, load;
5   input [AWIDTH-1:0] addr;
6   input [DWIDTH-1:0] d;
7   output [DWIDTH-1:0] q;
8   reg [DWIDTH-1:0] q;
9   reg [DWIDTH-1:0] mem [WORDS-1:0];
10
11  always @(posedge clk)
12    begin
13      if(load) mem[addr] <= d;
14      q <= mem[addr];
15    end
16
17  endmodule
```