

デュアルポートメモリ（ブロック RAM）

同期書き込み同期読み出しのデュアルポートメモリ。

ポート 1 は読み書き可．ポート 2 は読み出しのみ可

論理合成により，FPGA の ブロック RAM に変換される．

同時書き込みは読み出し優先 (read-first)．（同じ番地に読み出しと書き込みを同時に行うと，読み出しが先に行われる．）

パラメータ

パラメータ名	既定値	
DWIDTH	16	データのビット幅
AWIDTH	12	アドレスのビット幅
WORDS	4096	ワード数 ($=2^{\text{AWIDTH}}$)

入出力ポート

	ポート名	ビット数	
入力	clk		グローバルクロック
入力	load1	1	ポート 1 の書き込み許可 (1 のとき clk の立ち上がりで addr1 番地に d1 を書き込む)
入力	load2	1	ポート 2 の書き込み許可 (1 のとき clk の立ち上がりで addr2 番地に d2 を書き込む)
入力	addr1	AWIDTH	ポート 1 のアドレス指定
入力	addr2	AWIDTH	ポート 2 アドレス指定
入力	d1	DWIDTH	ポート 1 のデータ指定
入力	d2	DWIDTH	ポート 1 のデータ指定
出力	q1	DWIDTH	addr1 番地のデータ
出力	q2	DWIDTH	addr2 番地のデータ

ソースコード

```
module dpram(clk, load1, addr1, addr2, d1, q1, q2);
    parameter DWIDTH=16,AWIDTH=12,WORDS=4096;

    input clk,load1,load2;
    input [AWIDTH-1:0] addr1,addr2;
    input [DWIDTH-1:0] d1,d2;
    output [DWIDTH-1:0] q1,q2;
    reg [DWIDTH-1:0] q1,q2;
    reg [DWIDTH-1:0] mem [WORDS-1:0];

    always @(posedge clk)
```

```

begin
    if(load1) mem[addr1] <= d1;
    q1 <= mem[addr1];
end

always @(posedge clk)
begin
    if(load2) mem[addr2] <= d2;
    q2 <= mem[addr2];
end

integer i;
initial begin
    for(i=0;i<WORDS;i=i+1)
        mem[i]=0; // ここにメモリの初期化 (mem[12'h001]=16'h1234; など) を書く.
end

endmodule

```