スタック

パラメータ

パラメータ名	既定値	
N	8	スタックの要素数

入出力ポート

	ポート名	既定値	
入力	clk	1	グローバルクロック
入力	reset	1	グローバルリセット
入力	load	1	1 のとき clk の立ち上がりでスタックトップに d の値を書き込み
入力	push	1	1 のとき clk の立ち上がりでスタックをプッシュ
入力	pop	1	1 のとき clk の立ち上がりでスタックをポップ
入力	thru	1	1 のとき thru を qtop に出力
入力	d	16	スタックトップに書き込む値
入力	dthru	16	thru が 1 のとき dthru を qtop に出力
出力	qtop	16	thru が 1 のとき dthru，0 のときスタックトップの値
出力	qnext	16	スタックの 2 番目の値

PUSH 命令 (メモリから読み出したデータをスタックにプッシュ) を 1 クロックサイクルで実行するために，thru と dthru を導入.
メモリから読み出すのに 1 サイクル必要で，その読み出したデータをスタックトップに書き込むのにさらに 1 サイクル必要となる．そこでスタックに書き込むかわりに，メモリからの読み出しデータを dthru に入力し，それをスタックトップ qtop の値とする．つまり，フリップフロップ q[0] に書かれているスタックトップの値をスルーして，メモリの出力をスタックトップ qtop として出力する.

ソースコード

```
module stack(clk, reset, load, push, pop, thru, d, dthru, qtop, qnext);
  parameter N = 8;
```

```
  input clk, reset, load, push, pop, thru;
  input [15:0] d, dthru;
  output [15:0] qtop, qnext;
  reg [15:0] q [0:N-1];

  assign qtop = (thru ? dthru : q[0]);
  assign qnext = q[1];

  always @(posedge clk or negedge reset)
    if(!reset) q[0] <= 0;
    else if(load) q[0] <= d;
    else if(pop) q[0] <= q[1];
    else q[0] <= qtop;

  always @(posedge clk or negedge reset)
    if(!reset) q[1] <= 0;
    else if(push) q[1] <= qtop;
    else if(pop) q[1] <= q[2];

  integer i;
  always @(posedge clk or negedge reset)
    for(i=2;i<N-1;i=i+1)
      if(!reset) q[i] <= 0;
      else if(push) q[i] <= q[i-1];
      else if(pop) q[i] <= q[i+1];

  always @(posedge clk or negedge reset)
    if(!reset) q[N-1] <= 0;
    else if(push) q[N-1] <= q[N-2];

endmodule
```