afifo の仕様

2014 年 11 月 14 日

1　この非同期 FIFO について

非同期 FIFO のモジュールは afifo.v にあります。タイミング制約を満たせば、読み側と書き側とで別のクロックで動作します。

2　入出力ポート

入出力は以下の通りです。

```
module AFIFO
  #(parameter      DATA_WIDTH  = 32,
                   AFIFO_SIZE  = 4)

    (input                     RCLK, WCLK, RST_X,
     input                     enq, deq,
     input  [DATA_WIDTH-1:0]   data_in,
     output [DATA_WIDTH-1:0]   data_out,
     output                    empty,
     output                    full
     );
```

データサイズと FIFO の深さがパラメータで指定できます。デフォルトでは幅 32bit の深さ 16 です。RCLK が読み側のクロック、WCLK が書く側のクロックで、enq,deq は FIFO へのデータ書き込み、読み出し信号です。
この FIFO のメモリの現在の読み出しポインタを Rp, 次に書き込むポインタを Wp で保持しており、これらをグレイコードカウンタで制御しています。

3　write

FIFO への書き込みのタイミングは図 1 のようになります。

write 側が enq を 1 にして data_in に書き込みたいデータを入れると、full が立っていないときに限り、FIFO が enq が 1 になったのを posedge で確認し、そのときの data_in を FIFO のメモリに格納して同時に Wp を更新します。
FIFO の full 信号は Wp が Rp に追いついてしまったときに 1 になります。full が立つとデータの書き込みお

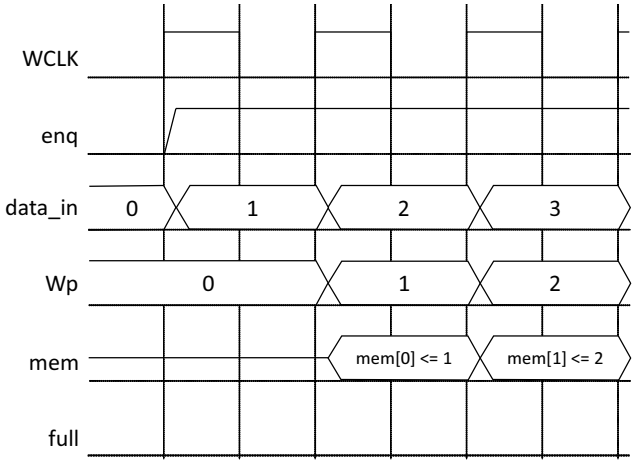

図 1　write のタイミング

よび Wp の更新を行いません。このときのタイミング図は図 2 となります。

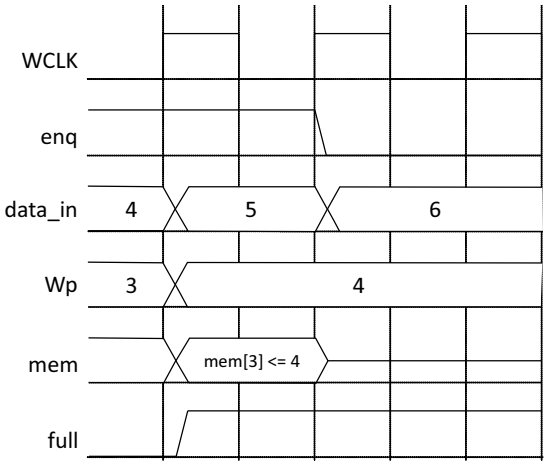

図 2　full のタイミング

full が立っているときの enq は無視されることになります。そのため、write 側は full が立っているのを確認できたら enq を下げ、書き込まれなかった 1 回分のデータを full が下がってからもう一度送る必要があります。

4　read

FIFO からのデータの読み出しのタイミングは図 3 のようになります。

data_out は常に現在の Rp のデータを出しています。empty が立っていないときには、deq は Rp の更新

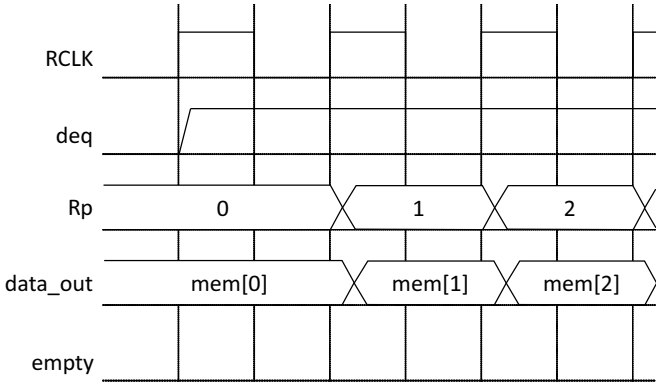

図 3　read のタイミング

信号になるので、FIFO が deq を確認し次第 Rp の値をインクリメントし、それに伴い data_out の値も更新されます。

FIFO の empty 信号は、Wp と Rp の値が同じになると 1 になります。empty が 立つと Rp の更新を行いません。このときのタイミング図は図 4 となります。empty が立っているときの deq は FIFO に無視されます。

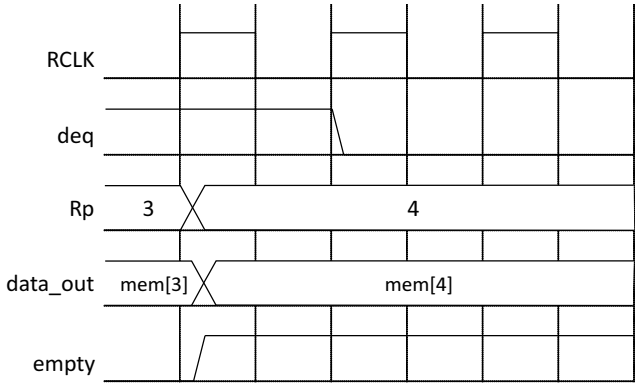

図 4　empty のタイミング

そのため、read 側は empty が立っているのを確認できたら deq を下げ、empty が下がってからもう一度データを読み出す必要があります。empty がでている間も data_out は現在の Rp のメモリの値を出し続けていますが、FIFO が空の状態での値となるので、正しいデータではありません。