AdcBulkMem

Marc Defossez
Sr. Staff Applications Engineer

Created: September 28, 2009
Modified: December 8, 2009

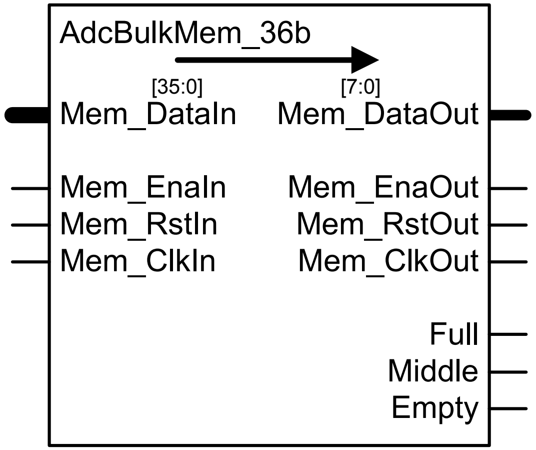
DISCLAIMER:

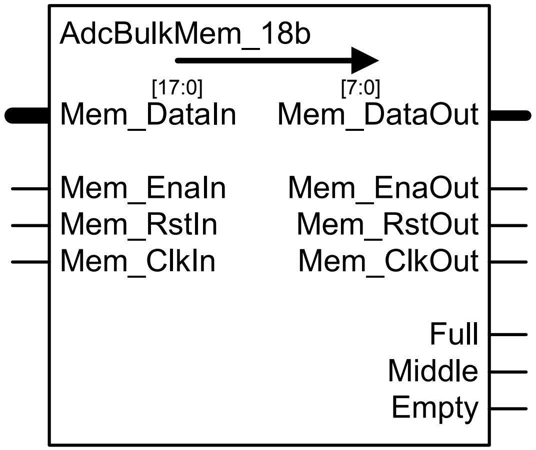
Contact: e-mail hotline@xilinx.com phone + 1 800 255 7778

XILINX

This page is intentionally left blank.

AdcBulkMem

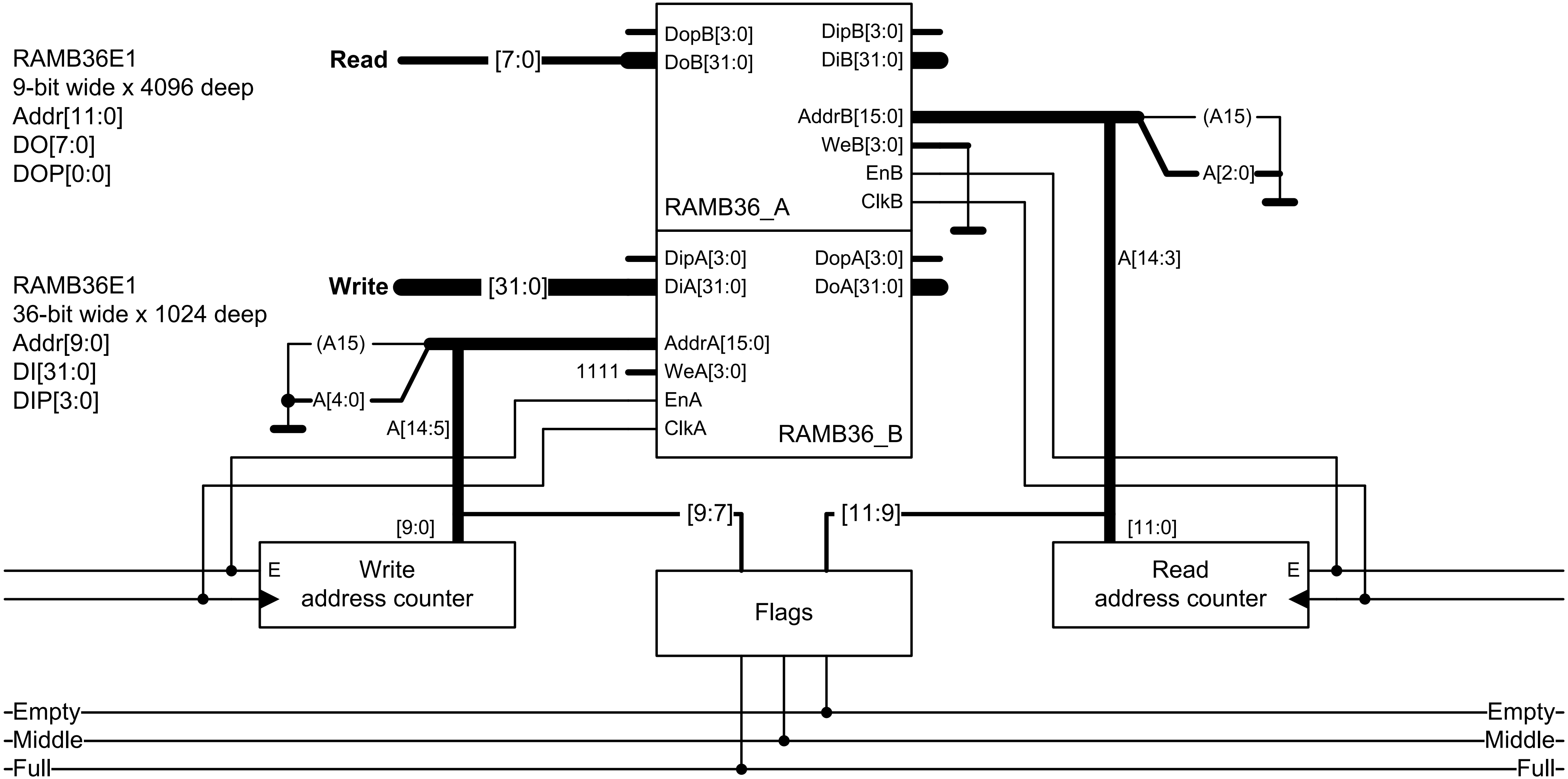




XILINX

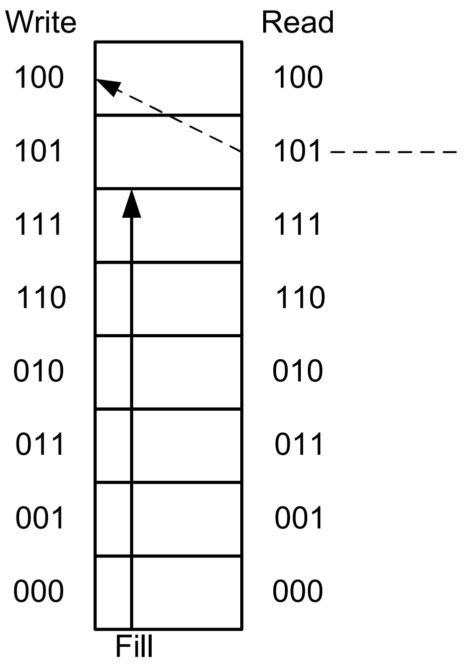
AdcBulkMem

The ADC writes into the Block-RAM and the processor reads from it.



XILINX

Flags – Gray code



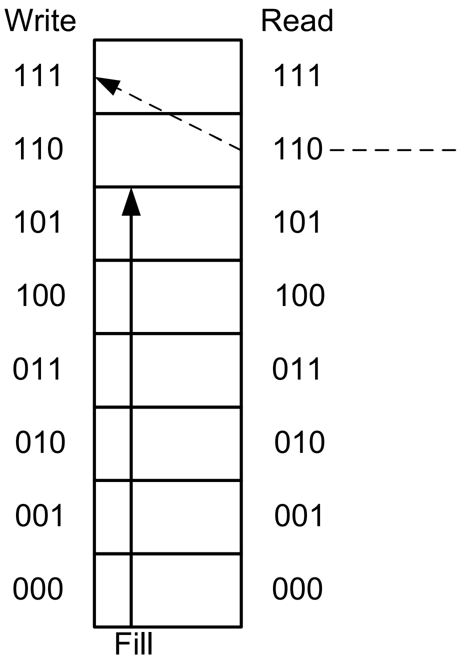
When the read pointer is right behind the write pointer, the FIFO is as good as empty.
When the write pointer is right behind the read pointer, the FIFO is nearly full.

EMPTY		FULL		Optional MIDDLE	
100	101	100	000	100	010
101	111	101	100	101	011
111	110	111	101	111	001
110	010	110	111	110	000
010	011	010	110	010	100
011	001	011	010	011	101
001	000	001	011	001	111
000	100	000	001	000	110
Write	Read	Write	Read	Write	Read

Assume that the write pointer runs faster than the read pointer, then:
The write pointer will reach “100” while the read pointer is still “000” the FULL flag goes high.
Writing must now be stopped until the read pointer reaches “101”, now the EMPTY flag can be set high and the FULL flag can be released.
Writing can continue from where it was stopped (Somewhere in the “100” region) and reading can be stopped. Writing goes over turn around in address space. When the write pointer reaches now “111 the FULL flag must be set again (Read pointer is at “101”). The read operation can start from where it last. When the read pointer reaches “110” the EMPTY flag can be set (reset the FULL flag).
And so on……..

Flags – Binary

When the counters are binary coded the flag logic should look like this;



When the read pointer is right behind the write pointer, the FIFO is as good as empty.
When the write pointer is right behind the read pointer, the FIFO is nearly full.

EMPTY

Write		Read
111		110
110		101
101		100
100	=	011
011		010
010		001
001		000
000		111

FULL

Write		Read
111		000
110		111
101		110
100	=	101
011		100
010		011
001		010
000		001

Optional MIDDLE

Write		Read
111		011
110		010
101		001
100	=	000
011		111
010		110
001		101
000		100

Assume that the write pointer runs faster than the read pointer, then:
The write pointer will reach “111” while the read pointer is still “000” the FULL flag goes high.
Writing must now be stopped until the read pointer reaches “110”, now the EMPTY flag can be set high and the FULL flag can be released.
Writing can continue from where it was stopped (Somewhere in the “111” region) and reading can be stopped. Writing goes over turn around in address space. When the write pointer reaches now “101 the FULL flag must be set again (Read pointer is at “110”). The read operation can start from where it last. When the read pointer reaches “100” the EMPTY flag can be set (reset the FULL flag).
And so on……..