I am going to modify the ROM of an old machine so that it can update custom firmware. Its system is on a mask ROM board.

The host is a spdc1016 soc, with an instruction set of 6502, 23-bit parallel address and an 8-bit AD multiplexer lines.

The pin definition of the BROM board is as shown below.

	Pin	Wen Quxing	Remark
	VSS		GND
	POWER	R23,R24,U7, EN	Slow change in resistance (10M)
	MC1		Address/data
	MC0		AH/AL R/W
	AD7	Nor29 DQ7	
	AD6	Nor28 DQ6	
	AD5	Nor27 DQ5	
	AD4	Nor26 DQ4	
	AD3	Nor25 DQ3	
	AD2	Nor23 DQ2	
	AD1	Nor22 DQ1	
	AD0	Nor21 DQ0	
	PRESZ		open circuit
	VDD		3.3v, 3.232V measured

There are three chips in parallel, each 4Mbitx8, with an address width of 16, and mapper is used to access different data.

The host spdc1016 uses common instructions such as STA to access 00..3F which are read/write registers, with gpio, timer, and mapper for different purposes. By the bank/vol/roa bits, the host can map the 4000...FFFF to different device memory cells.

The mapper host and the memory chip are switched synchronously, when the code writes to the registers in the host, the adbus will also have the write timing, so that the busrom can switch its own mapping.

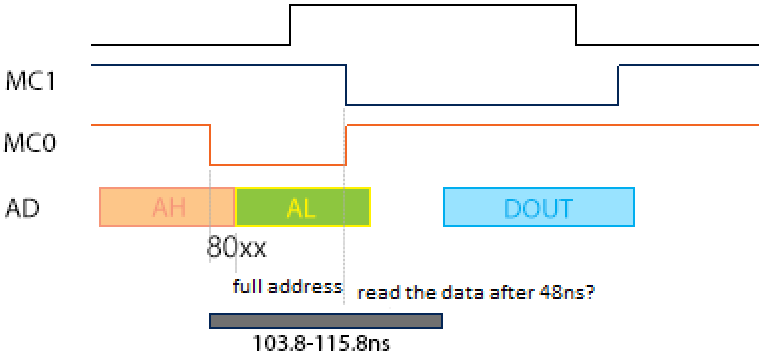
For the three busrom (B1/B2/B3), B1 responds to all vol 0..7F banks at 4000..BFFF and all bios banks at C000..DFFF, E000..FFFF responds to fixed mapping, and B2/B3 at 4000..BFFF responses to vol0/vol1 of 80..FF bank.

bank[7:0] is located at register address 00[7:0], vol at 0D[1:0], roa at 0A[7], and bios bank at 0A[3:0]. These addresses are stored in reg when written in verilog, and then the 24-bit physical address is dynamically generated when the host reads the original ROM.

The following is the mc1/mc0 signal, where the high range of mc1 indicates the address of the ad bus, and the low range of the ad bus is the data.

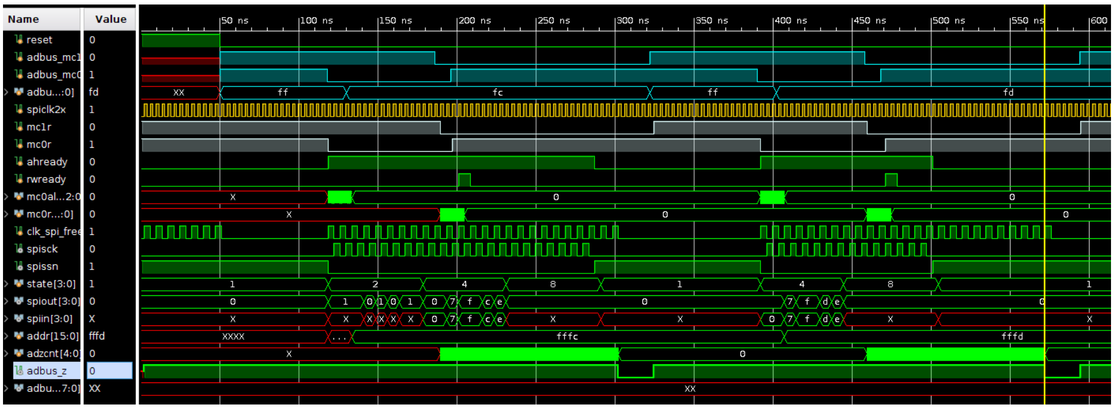
The mc0 signal indicates whether the transmission is AH or AL in the address phase, and whether it is a read or write operation in the data phase.

The cpu access, because of the phase difference between mc1 and the cpu clock, can be regarded as single-cycle, that is, in the same cpu clock cycle, the address is given and the data is read, and cannot be delayed until the next cycle (because the bus may output the address immediately).



The cpuclk, mc1/mc0/ad should ideally be read with the timing as above. The frequency is 3.6864M, because the qspi flash is used to store the firmware, the data and instruction consumption becomes a big bottleneck.

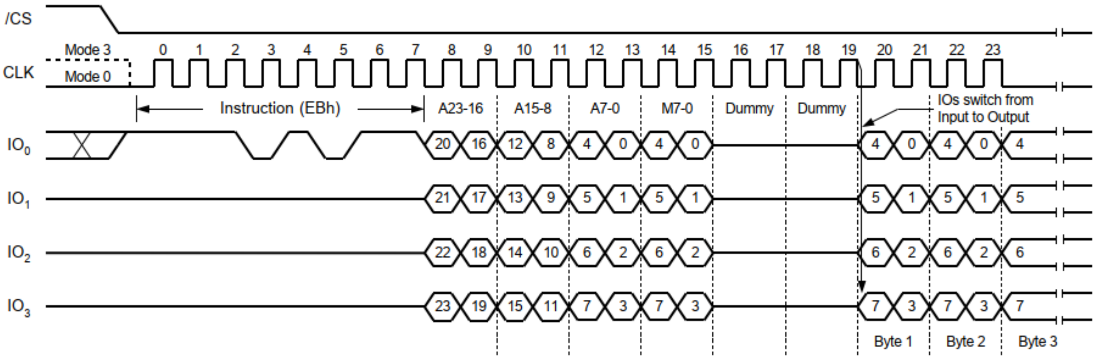
So my optimization is that, once the MC0 drop is captured, I can start the spi operation, and wait until the lower 8 bits are sent by the 12th/4th clocks, when the AD period is located in the AL interval (green in the figure).



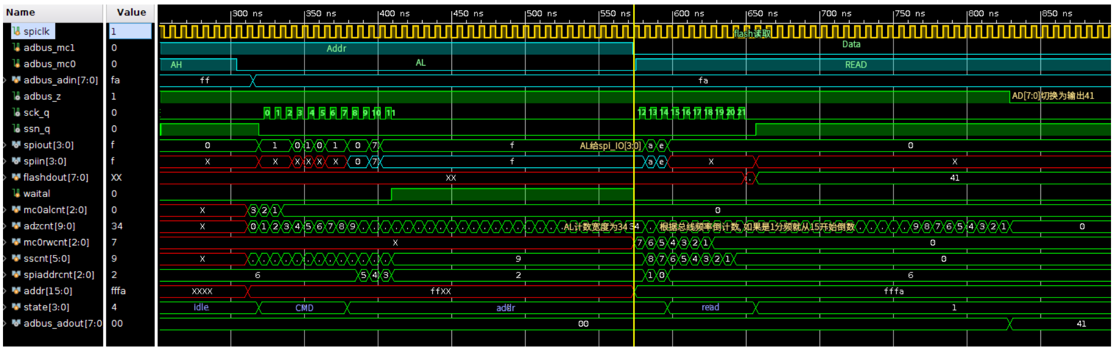
Name	Value
reset	0
adbus_mc1	0
adbus_mc0	1
adbu...:0]	fd
spiclk2x	1
mc1r	0
mc0r	1
ahready	0
rwready	0
mc0al...2:0	0
mc0r...:0]	0
clk_spi_free	1
spisck	0
spissn	1
state[3:0]	1
spiout[3:0]	0
spiin[3:0]	X
addr[15:0]	fffd
adzcnt[4:0]	0
adbus_z	0
adbu...7:0]	XX

I verify the waveform in the simulator. MC initialization is AH state, and AD outputs FFFC, holding for 12ns after MC0 falling edge; MC0 rising is 10ns later than MC0 falling.

The simulation waveform shows that the host clock is half a clock ahead of the clock seen by the device; at first, the device reads the data at the last falling edge of the dummy clock, corresponding to the 19/20 rising edge of clk_spi_free, but because of the delay, the device has to read on the rising edge of the 20/21 clock. The single edge also helps with timing optimization.



The modified simulation waveform is shown below:



Name	Value
spiclk	1
adbus_mc1	0
adbus_mc0	0
adbus_adin[7:0]	fa
adbus_z	1
sck_q	0
ssn_q	0
spiout[3:0]	f
spiin[3:0]	f
flashdout[7:0]	XX
waital	0
mc0alcnt[2:0]	0
adzcnt[9:0]	34
mc0rwcnt[2:0]	7
sscnt[5:0]	9
spiaddrcnt[2:0]	2
addr[15:0]	fffa
state[3:0]	4
adbus_adout[7:0	00

It can be seen that AL is forwarded from adbus to spi only after MC1 drops. For bus 4 division, slow reads are also selected at the center of the Data interval to avoid possible level conflicts due to excessive slowness.

Attachment: Development Board Pin Assignments

Signal	Pin	Color
CS	30	purple
SCK	29	Green
IO0	35	Blue
IO1	34	Gray
IO2	32	White
IO3	31	Yellow
MC1	47	Brown

MC0	48	Red
AD0	39	Black
AD1	40	White
AD2	41	Gray
AD3	42	Purple
AD4	43	Blue
AD5	44	Green
AD6	45	Yellow
AD7	46	Orange