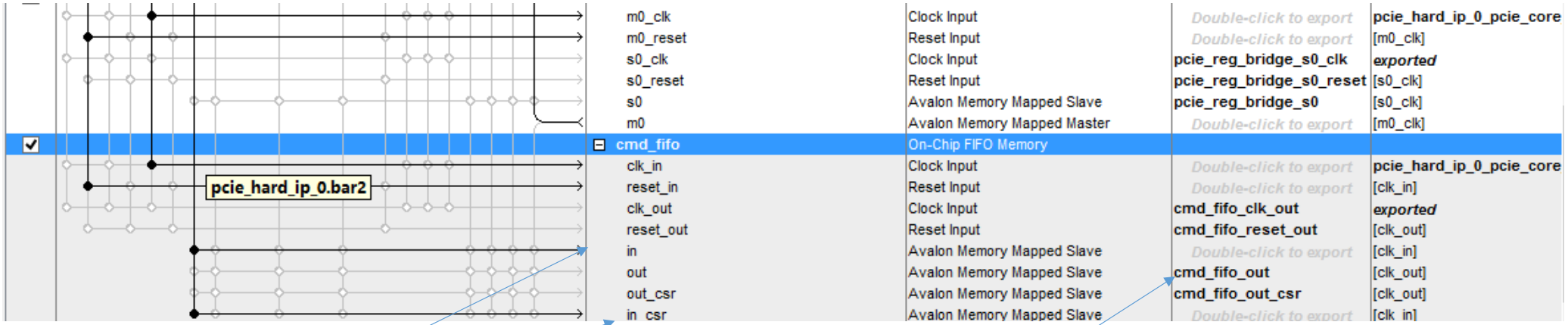


I add a FIFO memory on the basis of Altera PCIR DDR3 reference design
IN & IN_CSR are connected to Bar0/1

OUT connect to user logic (Avalon-master)

Table 16–4. FIFO Status Field Descriptions (Part 1 of 2)

Field	Type	Description
fill_level	RO	The instantaneous fill level of the FIFO, provided in units of symbols for a FIFO with an Avalon-ST FIFO and words for an Avalon-MM FIFO.
i_status	RO	A 6-bit register that shows the FIFO's instantaneous status. See Table 16–5 for the meaning of each bit field.
event	RW1C	A 6-bit register with exactly the same fields as i_status. When a bit in the i_status register is set, the same bit in the event register is set. The bit in the event register is only cleared when software writes a 1 to that bit.
interruptenable	RW	A 6-bit interrupt enable register with exactly the same fields as the event and i_status registers. When a bit in the event register transitions from a 0 to a 1, and the corresponding bit in interruptenable is set, the master Is interrupted.
almostfull	RW	A threshold level used for interrupts and status. Can be written by the Avalon-MM status master at any time. The default threshold value for DCFIFO is Depth-4. The default threshold value for SCFIFO is Depth-1. The valid range of the threshold value is from 1 to the default. 1 is used when attempting to write a value smaller than 1. The default is used when attempting to write a value larger than the default.
almostempty	RW	A threshold level used for interrupts and status. Can be written by the Avalon-MM status master at any time. The default threshold value for DCFIFO is 1. The default threshold value for SCFIFO is 1. The valid range of the threshold value is from 1 to the maximum allowable almostfull threshold. 1 is used when attempting to write a value smaller than 1. The maximum allowable is used when attempting to write a value larger than the maximum allowable.

PC read the REG is right(I setup the FIFO depth is 16, so read them before writing FIFO)
The status REG value is "0000000A"

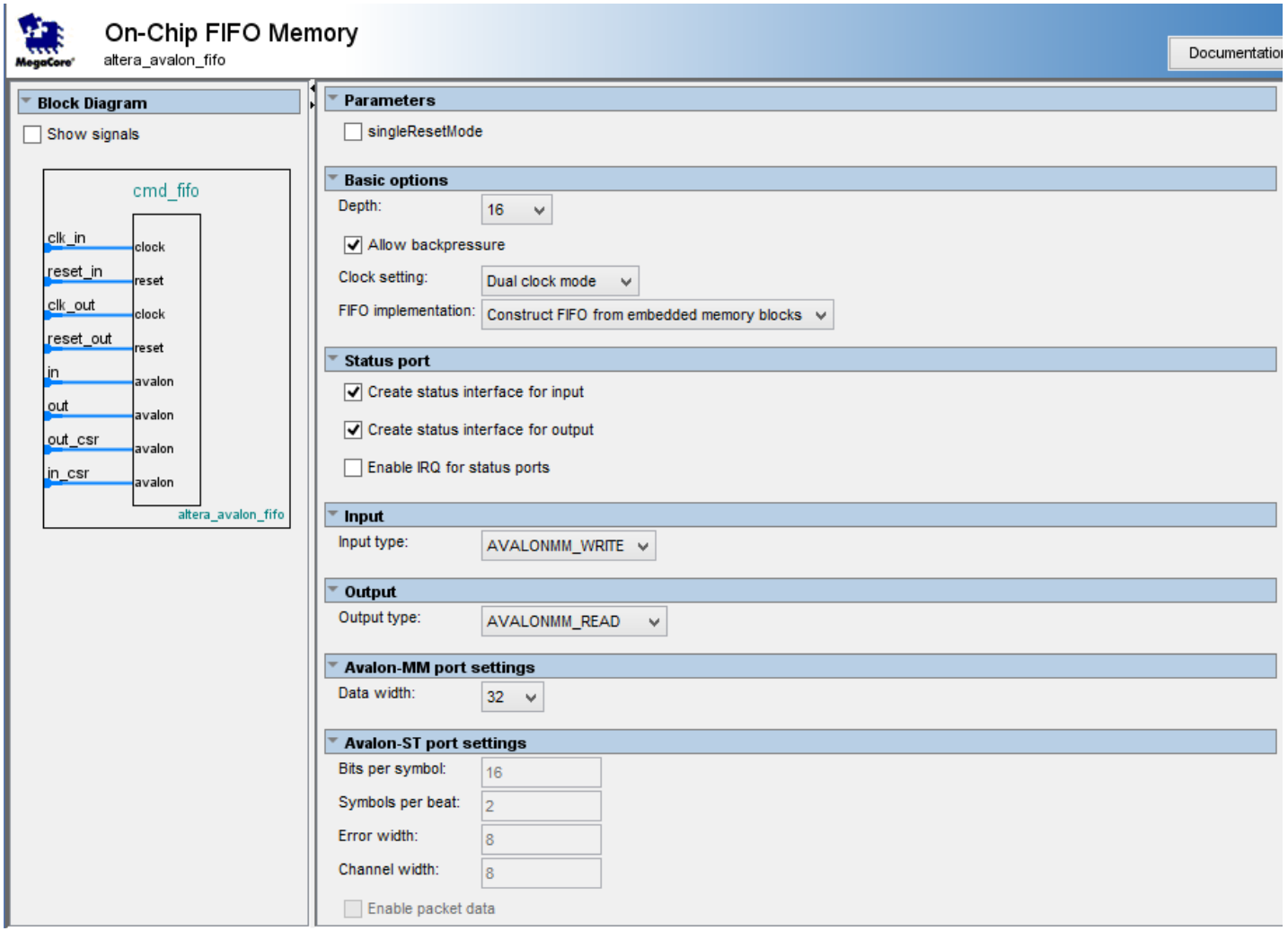
The almostfull REG value is "0000000C"

It seems meet the REG description

Either read or write these REGs, PC host is normal

But, when I write data to FIFO core, it will result in PC hung up.(address is 0x500_0000, repeat write data 6 times)

below picture is fifo setup.



Qsys address map setup

System Contents | Address Map | Clock Settings | Project Settings | Instance Parameters | System Inspector | HDL Example | Generation

	pcie_hard_ip_0.bar1_0	pcie_hard_ip_0.bar2	dma_read_master_0.Data_Read_Master	dma_write_master_0.Data_Write_Master
altmemddr_0.s1	0x0800_0000 - 0x0fff_ffff		0x0800_0000 - 0x0fff_ffff	0x0800_0000 - 0x0fff_ffff
pcie_hard_ip_0.txs			0x0000_0000 - 0x00ff_ffff	0x0000_0000 - 0x00ff_ffff
pcie_hard_ip_0.cra		0x0000_0000 - 0x0000_3fff		
onchip_memory2_0.s1	0x0700_0000 - 0x0703_ffff		0x0700_0000 - 0x0703_ffff	0x0700_0000 - 0x0703_ffff
onchip_memory2_0.s2				
modular_sgdma_dispatcher_0.CSR		0x0600_0000 - 0x0600_001f		
modular_sgdma_dispatcher_0.Descript...		0x0600_0020 - 0x0600_002f		
half_rate_bridge_0.s1				
read_cra_bridge.s0				
CSR_bridge.s0				
PCIE_REG_Bridge.s0				
cmd_fifo.in	0x0500_0000 - 0x0500_0003			
cmd_fifo.out				
cmd_fifo.out_csr				
cmd_fifo.in_csr	0x0500_0020 - 0x0500_003f			
altmemddr_0.s1 via half_rate_bridge_0				
pcie_hard_ip_0.txs via PCIE_REG_Bridge				
modular_sgdma_dispatcher_0.CSR via ...				
pcie_hard_ip_0.cra via read_cra_bridge				
modular_sgdma_dispatcher_0.Descript...				

VS /PC test program:

```
unsigned int FIFO_addr_SCR_REG = 0x05000020;

printf("\n fifo status read befeore control \n");
for(int i = 0; i < 8; i++){
    unsigned int ret = OnRCSlaveRead(hDev, 0, FIFO_addr_SCR_REG + i*4);
    printf("%x fifo_ : %x\n", FIFO_addr_SCR_REG + i*4, ret);
}

OnRCSlaveWrite(hDev, 0, FIFO_addr_SCR_REG+0x10 ,0x6);//almost full
OnRCSlaveWrite(hDev, 0, FIFO_addr_SCR_REG+0x14 ,0x3);//almost empty
OnRCSlaveWrite(hDev, 0, FIFO_addr_SCR_REG+0x8 ,0x0);//event

printf("\n FIFO status read after control \n");
for(int i = 0; i < 8; i++){
    unsigned int ret = OnRCSlaveRead(hDev, 0, FIFO_addr_SCR_REG + i*4);
    printf("%x fifo_CSR : %x\n", FIFO_addr_SCR_REG + i*4, ret);
}


unsigned int FIFO_addr_data = 0x05000000;
printf("\n Alex fifo data write \n");
for(int i = 0; i < 6; i++){
    // OnRCSlaveWrite(hDev, 0,0x05000000, 0x12345678+i);
}
```

Read status REG OK

write status REG sometimes OK , sometimes failed, don't know the reasom

Read status REG again!

When write data to fifo core, PC can hung up.
I am sure that This green line result in PC hung up or PCIE lost

Would you please help me to fix this issue?.

Thanks