DSP VLSI Systems
Homework (X)

SoC and DSP Architecture

Deadline: You don’t need to deliver this homework.

1. Figure 1 shows a simplified system with DMA and two memory blocks. The memory mapped I/O for the DMA module is shown in Table 1.

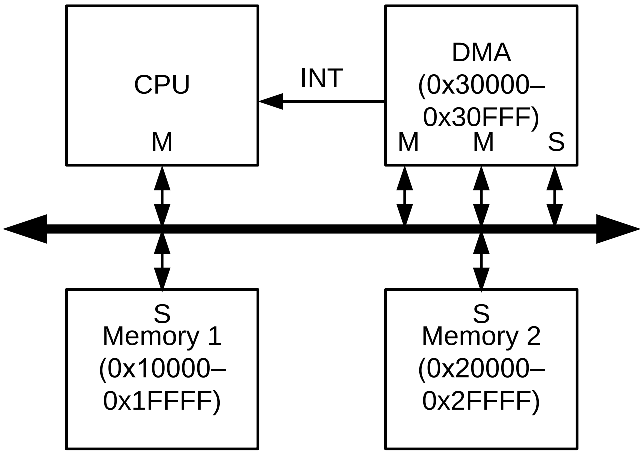


Fig. 1. Simplified system with DMA and two memory blocks.

Table 1. Memory-mapped I/O register of the DMA module.

Offset	Signal	Size	R/W	Description	Reset Value
0x00	Start	[0]	W	Start signal. 0: no operation. 1: start.	0x0
0x04	Status	[0]	R	Status register. 0: idle. 1: busy.	0x0
0x08	Source address	[31:0]	W	Source address of the data transferring	0x0
0x0C	Destination address	[31:0]	W	Destination address of the data transferring	0x0
0x10	Size	[5:0]	W	Size (number of words) of the data transferring.	0x0

If we want to move 32 words from 0x10000 to 0x20000, please design the pseudo-program with the ISR.
(Hint 1: please compose the program by use of the memory read/write functions: *read(address, &data)* and *write(address, data)*.
Hint 2: don't use polling, design with polling can only get half of scores)

2. Figure 2 shows a classical 3-tap FIR filter in direct form. Please design the wrapper for the FIR filter as an FIR accelerator IP in an SoC, as shown in Fig. 3. Here are the design guidelines:
1) Connect the FIR with an input FIFO and output FIFO.
2) The bus is an AHB-like bus system.
3) The accelerator has interrupt capability.
4) The IP has one AHB master for data read/write and one AHB slave interface for control register.

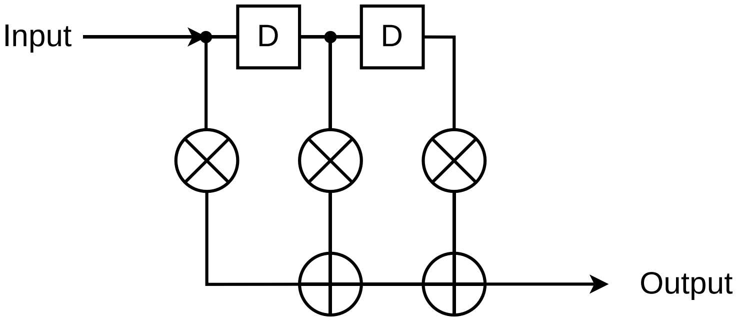


Fig. 2. A 3-tag FIR filter in direct form.

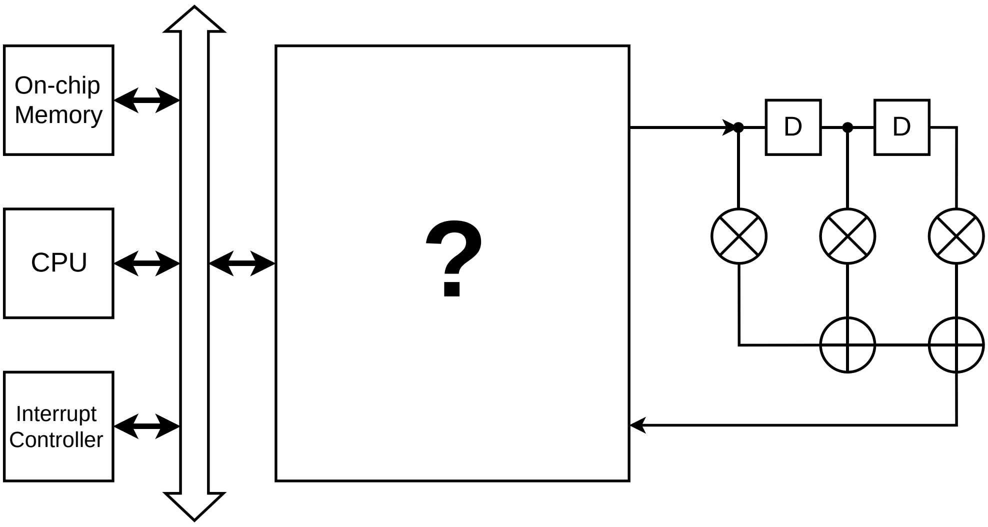


Fig. 3. The FIR accelerator IP in an SoC.

1.

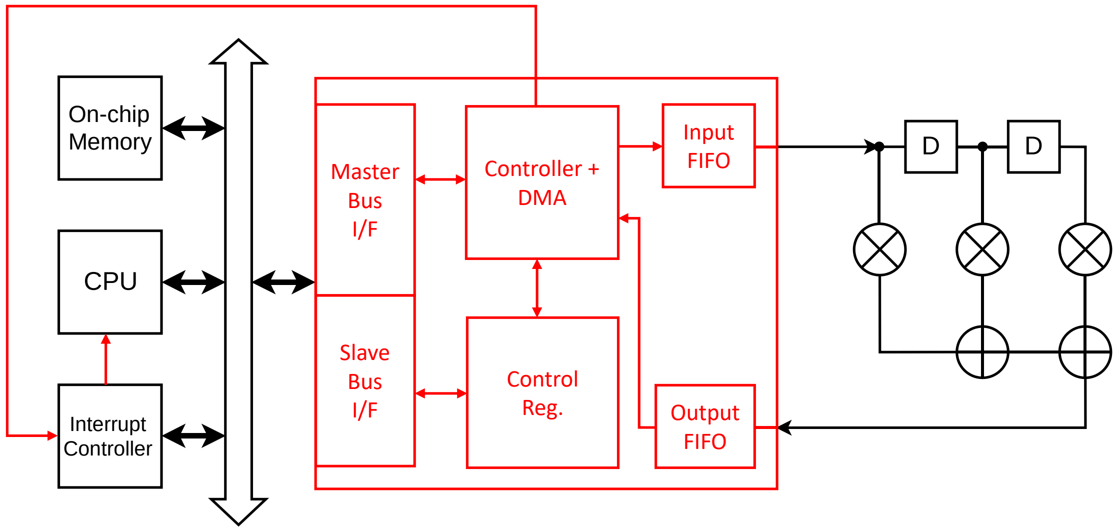
```
int source_addr = 0x10000;
int dest_addr=0x20000;
void call_DMA( );
void DMA_ISR( );

main()
{
  source_addr = 0x10000;
  dest_addr=0x20000;
  call_DMA( );
  ……
}

call_DMA()
{
    while(1)
    {
        read(0x30004, &status);
        if(status==0) break;
    }
    write(0x30000, 0); //stop
    write(0x30008, source_addr);
    write(0x3000C, dest_addr);
    write(0x30010, 32); //size = 32
    write(0x30000, 1); //start
}

DMA_ISR()
{
    read(0x30004, &status);
    if(status!=0) return;
    else write(0x30000, 0);
    return;
}
```

2.


On-chip Memory
CPU
Interrupt Controller
Master Bus I/F
Slave Bus I/F
Controller + DMA
Control Reg.
Input FIFO
Output FIFO
D
D