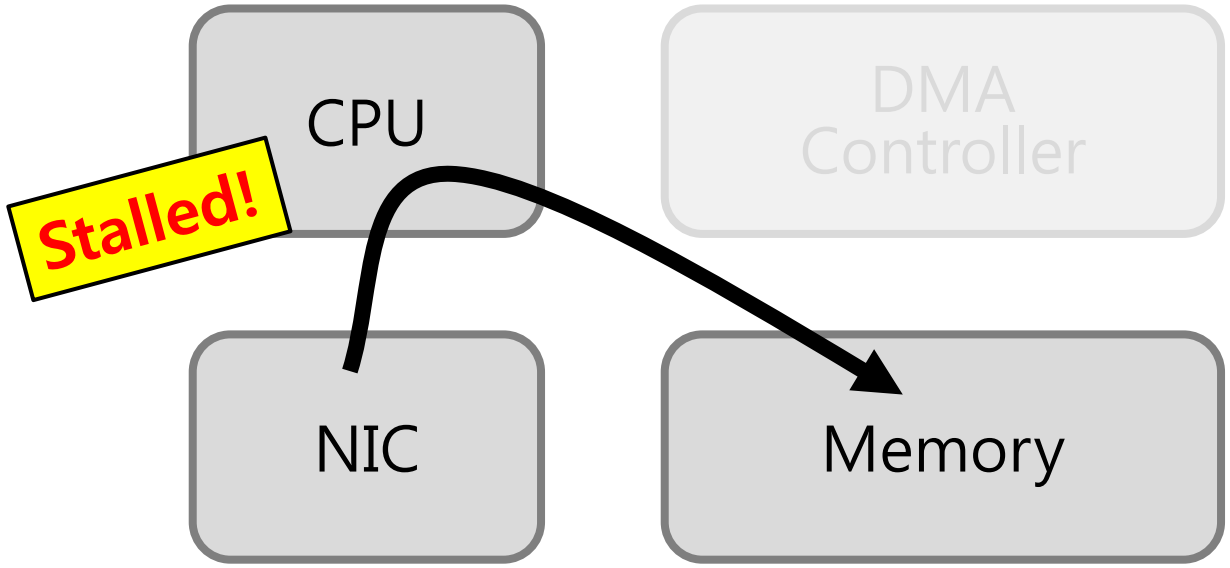
CSED311 Lab7:
Direct Memory Access

Byeong-Hoon So
sbh4728@postech.ac.kr

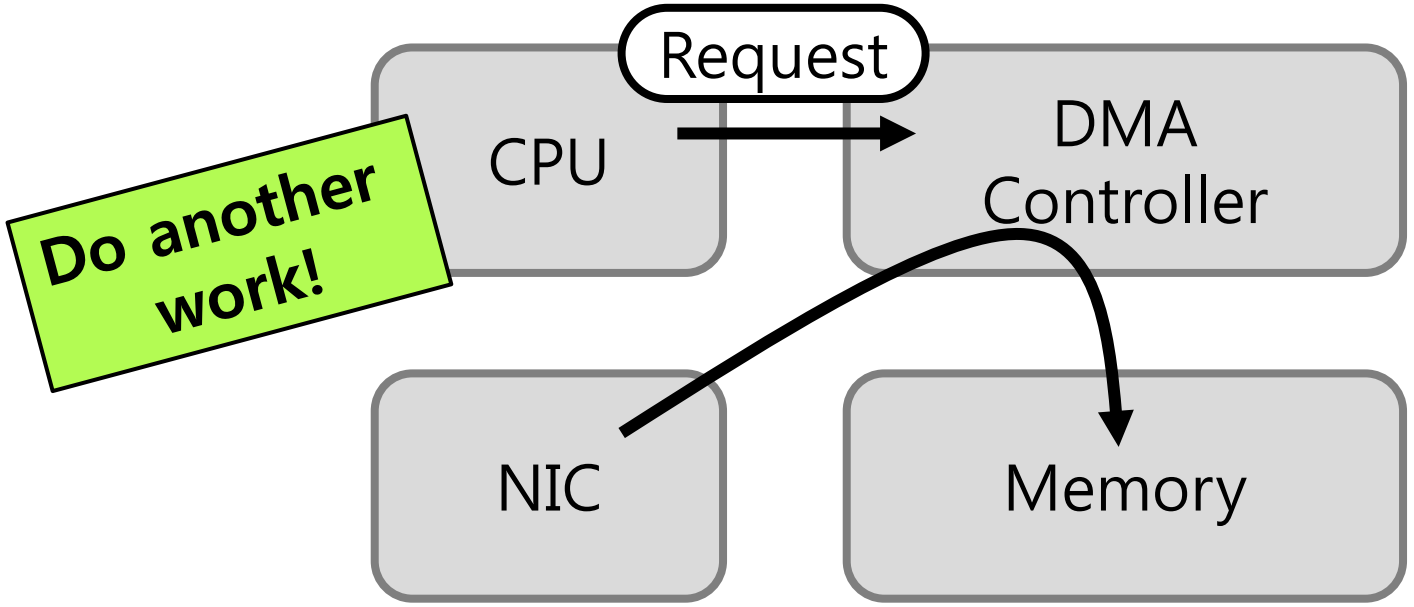
POSTECH

Direct Memory Access



- CPU must be stalled on transferring data from I/O devices to the memory.
  - To transfer data, CPU must read data from the I/O device and write them to the memory.
  - During the process, CPU cannot do useful work.

Direct Memory Access



- By letting the DMA controller load data, CPU do another work while transferring the data.
  - CPU requests the DMA controller to load data.
  - The DMA controller then loads the data on behalf of the CPU.
  - CPU can do another work while the DMA controller transfers data.

Experiment Scenario

- Let's simulate an external I/O device.
  - An external device that has at least 12 words of data.
  - The device notifies to the CPU that it has some data to transfer.
  - The CPU then transfers the 12 words of data from the device to the memory, using the DMA controller.

- Our memory model has 4 words of bandwidth.
  - Then, 12 words of data requires 3 (=12/4) memory transactions.
    i.e., 4 word write x 3 times

Experiment Scenario

- Let's simulate an external I/O device.
  - An external device that has at least 12 words of data.
  - The device notifies to the CPU that it has some data to transfer.
  - The CPU then transfers the 12 words of data from the device to the memory, using the DMA controller.
- Our memory model has 4 words of bandwidth
  - Then, 12 words of data requires 3 (=12/4) memory transactions.
    i.e., 4 word write x 3 times

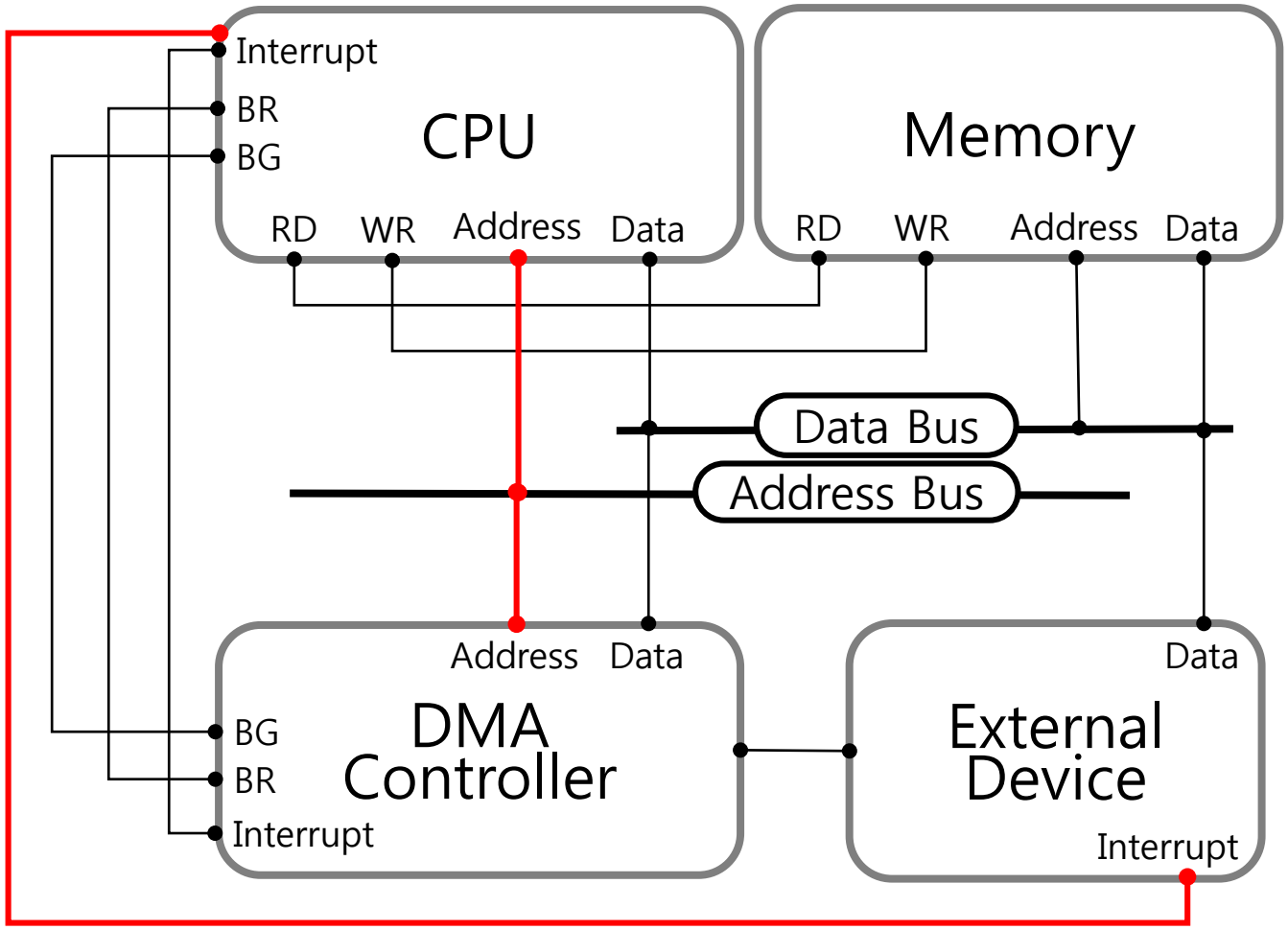
Need to write a short Verilog code for this device!

And this DMA controller too!

In testbench …

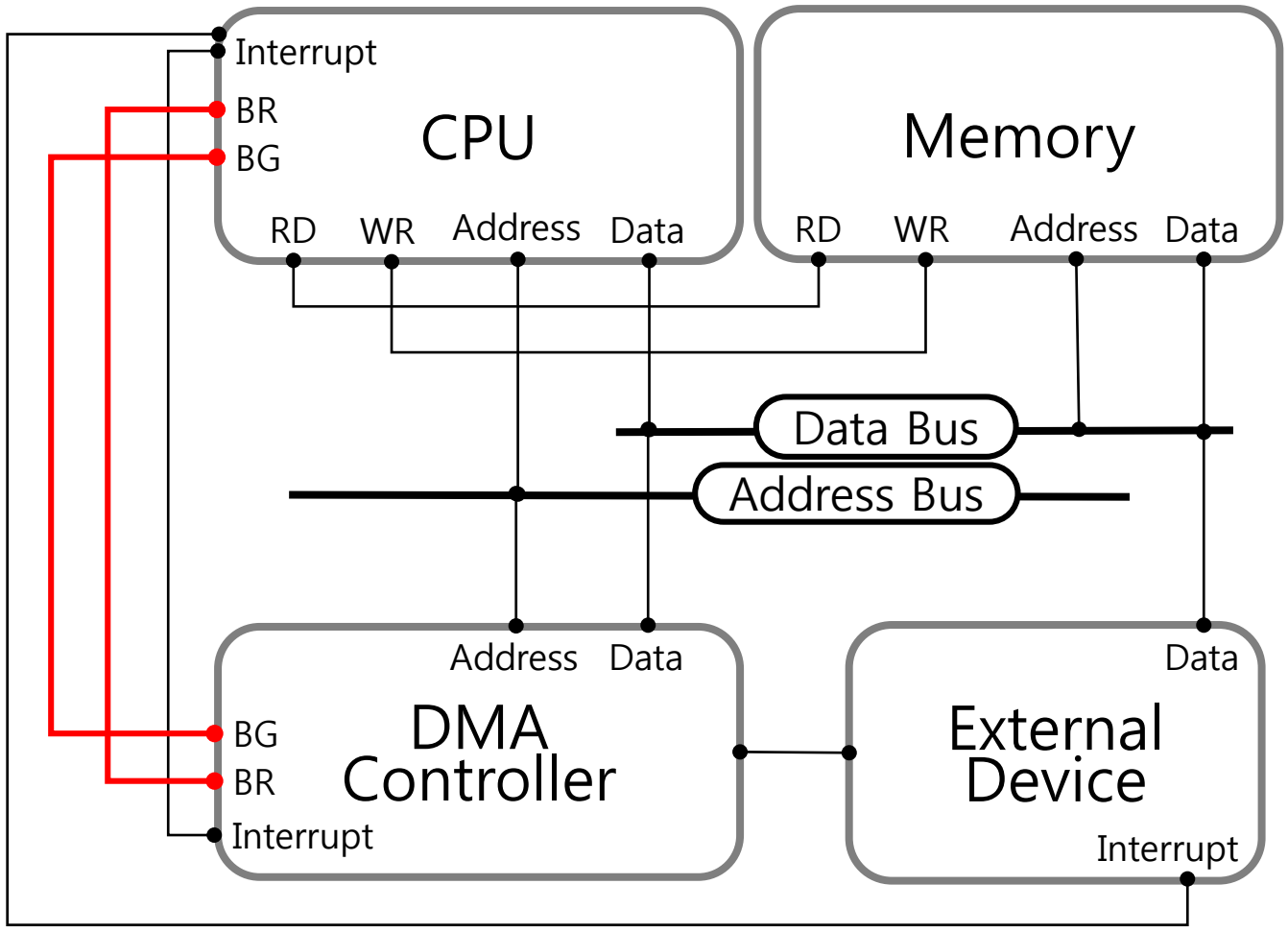
```
cpu UUT ( ... );
Memory NUUT ( ... );
DMA_controller DMAC ( ... );
external_device ED ( ... );
```

Detailed Scenario



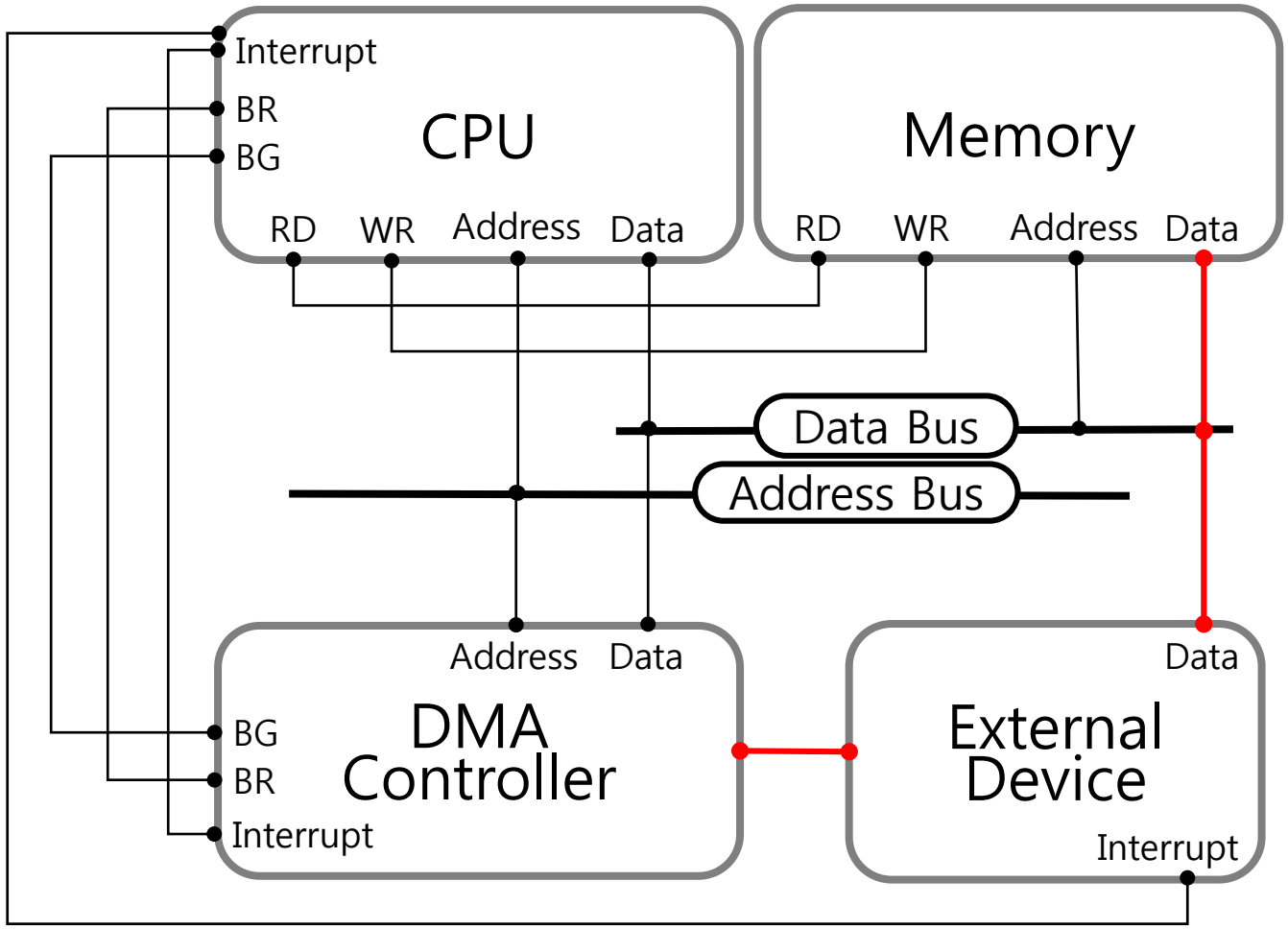
1. An external device sends an interrupt to a CPU.
   - You should design an interface for the interrupt.
   - CPU should always be ready for the interrupt.
2. The CPU tells a command (length, address) to a DMA controller.
   - length : the length of data. **12 words (fixed)**
   - address : the target memory address

Detailed Scenario



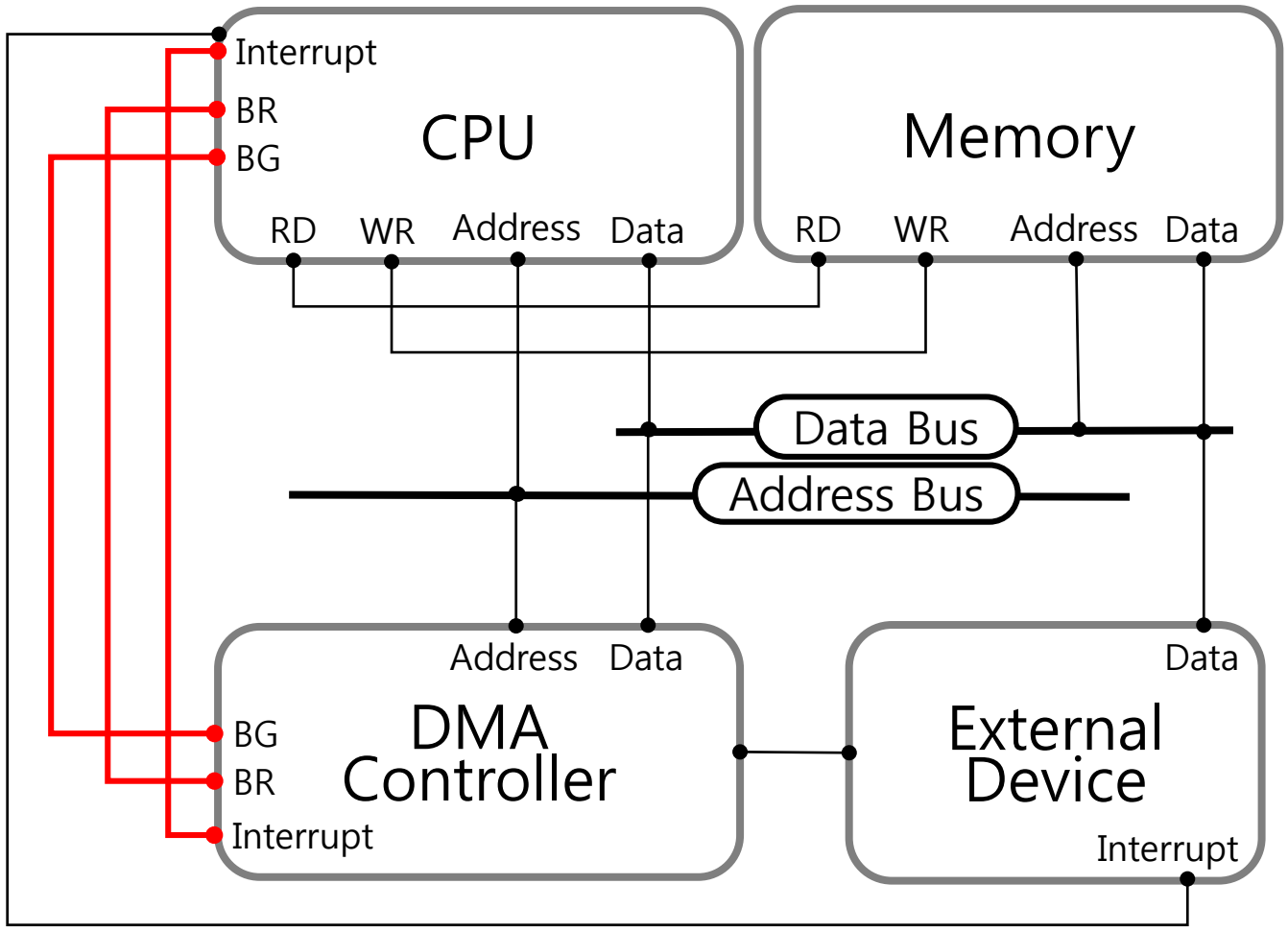
3. The DMA controller raises a BR(Bus Request) signal.
4. The CPU receives the signal, then it blocks its usage of address bus, data bus, and read/write line.
5. The CPU raises a BG(Bus Granted) signal.

Detailed Scenario



6. The DMA controller receives the signal. The external device writes 12 words of data on the designated memory address.
7. At that time, CPU should run with only its cache. It should be blocked when a cache miss is occurred.
   - Because it cannot access the memory now.

Detailed Scenario



8. When the DMA controller finish its work, it clears the BR signal.
9. The CPU clears the BG signal, and enables the usage of memory buses.
10. The DMA controller raises an interrupt
11. The CPU handle the interrupt

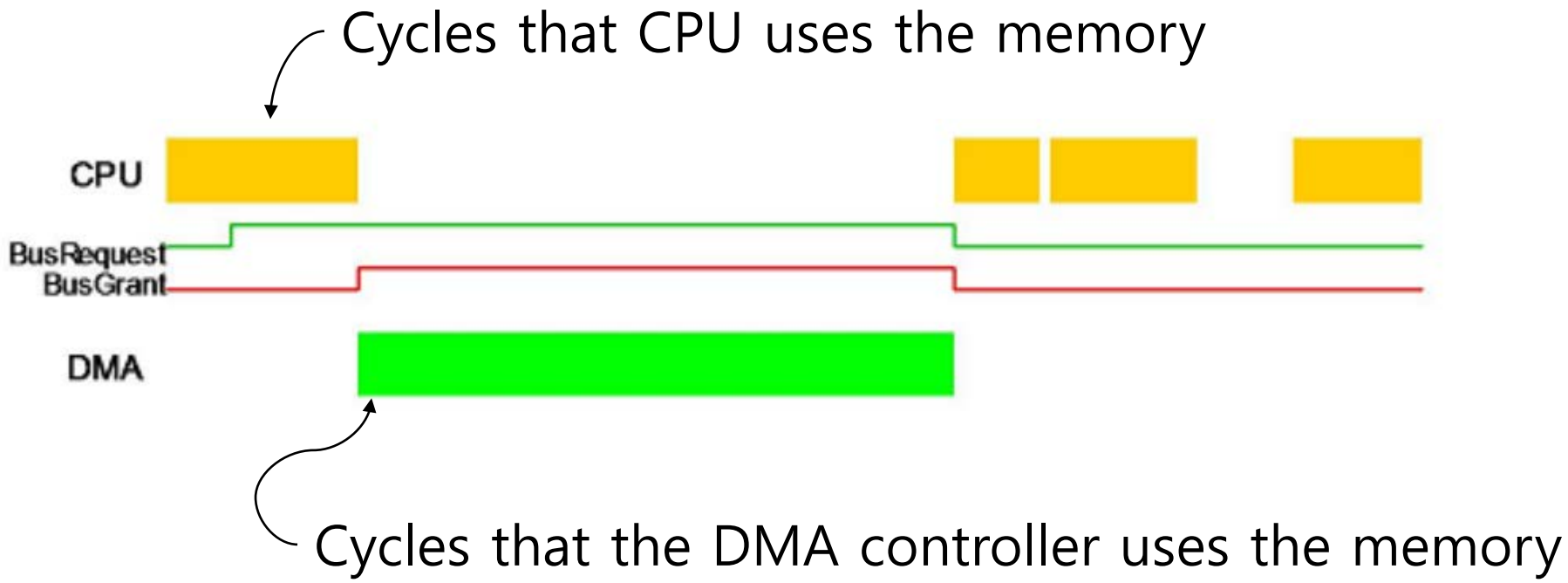
What to Implement: CPU

- CPU Part
  - A simple interrupt handling logic
    - Two kinds of interrupt : DMA begin, DMA end
    - CPU should be halted and process the interrupt when it occurs.
  - A bus arbitrating logic
    - When the DMA controller requests the memory buses, it should be granted as soon as possible.
    - When the DMA controller uses the memory buses exclusively, the CPU should not use them.
    - When the DMA controller releases the memory buses, it should enable the usage of them again.

What to Implement: Etc.

- External Device Part
  - It should waits few hundreds of clock cycles, then raise an interrupt. This interrupt means that the device has at least 12 words of data to send.
  - When the DMA controller tells the offset of data (0, 4 or 8), writes the proper data on the data bus.
- DMA Controller Part
  - When the CPU tells a DMA command, it should receive the command and try to grab the memory buses exclusively.
  - It should designate the memory address, and make the external device write the data.
  - It should tells an interrupt to the CPU at the end of DMA.

Working Model (Waveform)



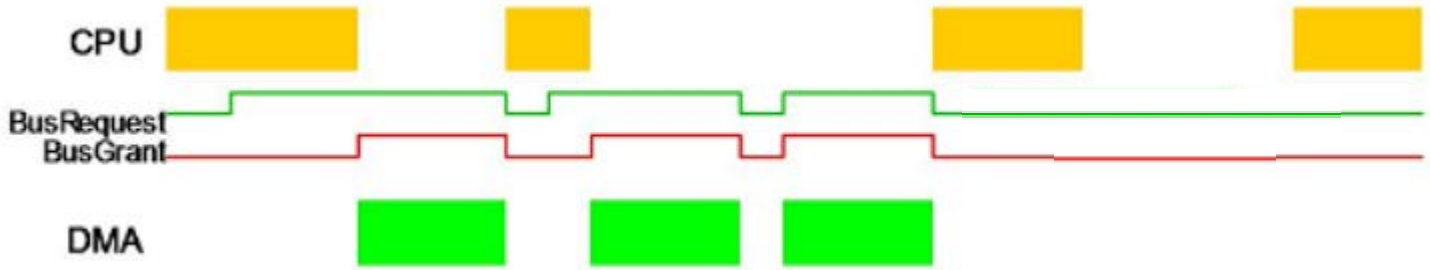
Grading Criteria

- Your CPU must be still functional!
  - Your CPU should pass every tests on the previous testbench.
  - Its functional correctness must not be affected by the DMA.
- Your CPU must handle external signals and interrupts properly.
- Your CPU must be working while the DMA is working, except when it has a cache miss.
- After the “End-of-DMA” interrupt, the device’s data should be read from the designated memory address.

You should prove these by a waveform or additional WWD tests.

Extra Credit (1)

- After the DMA controller sends 4 words, it releases the bus and grabs them again.
- If the CPU needs the memory, it will use it for a time.
- Otherwise, the DMA controller will retake the bus as soon as possible.



Extra Credit (2)

- The cache line may be out-of-sync just after the memory receives data from the external device.
  - ex) cache line for the address 0x32 = 0xAA
    received data written in the address 0x32 = 0xBB
- The cache line corresponding to the received address should be invalidated.

Q&A

Please write on LMS Q&A board