Question 1: HW-4 (2018)

Three 3-wire DMACs ($DMAC_1$, $DMAC_2$, and $DMAC_3$) are connected to the 68000-like processor over a bus arbiter (see lecture notes slide 5.23).
The CPU has the following properties.

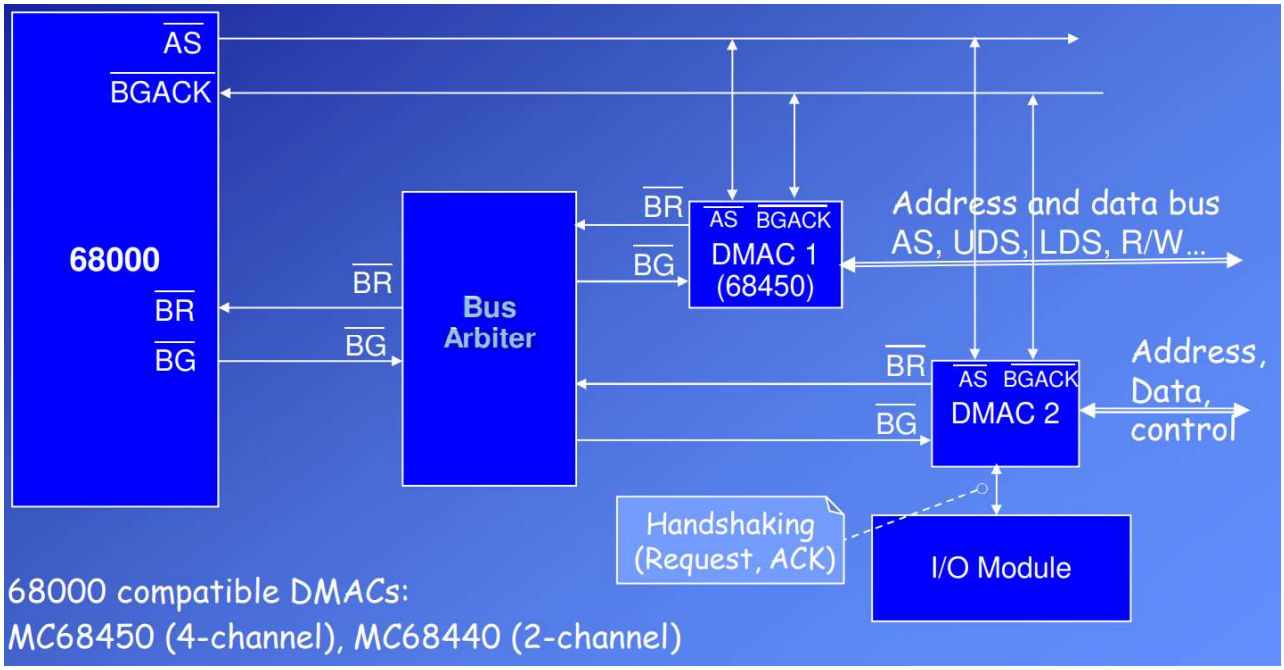
- No Pipelining
- 16-bit Data Bus
- 4 cycles:
  - Instruction Fetch
  - Operand Fetch
  - Execution
  - Operand Write
- Only execution cycle does not need memory.

The order of precedence is $DMAC_1 > DMAC_2 > DMAC_3$.

a) Construct the truth table for the bus arbiter. BR_i represents the BR output of the $DMAC_i$, and BG_i represents the BG input of the $DMAC_i$.

b) Write the minimized logic expressions for the outputs of the bus arbiter.

c) $DMAC_1$ and $DMAC_3$ are in the cycle-stealing mode, and $DMAC_2$ is in the burst mode. All DMACs are programmed to transfer 10 words. Assume that $DMAC_1$, $DMAC_2$ and $DMAC_3$ issue bus requests at the same time as the processor is in the instruction fetch cycle. Write step by step operations performed by the DMACs and the CPU from the beginning of the instruction fetch until the end of the completion of the first instruction.



Answer

a) All signals are zero-active. (30 Points)

BR1	BR2	BR3	BG	BR	BG1	BG2	BG3
0	X	X	0	0	0	1	1
0	X	X	1	0	1	1	1
1	0	X	0	0	1	0	1
1	0	X	1	0	1	1	1
1	1	0	0	0	1	1	0
1	1	0	1	0	1	1	1
1	1	1	1	1	1	1	1

b) (20 Points)
BR= BR1.BR2.BR3
BG1 = BR1 + BG
BG2 = BR1' + BR2 + BG
BG3 = BR1' + BR2' + BR3 + BG

c) (50 Points)

Cycle	CPU	1st DMAC	2nd DMAC	3rd DMAC
1	Fetch			
2		1st Word		
3			1st Word	
4			2nd Word	
5			3rd Word	
6			4th Word	
7			5th Word	
8			6th Word	
9			7th Word	
10			8th Word	
11			9th Word	
12			10th Word	
13		2nd Word		
14				1st Word
15		3rd Word		
16				2nd Word
17		4th Word		
18				3rd Word
19		5th Word		
20				4th Word
21		6th Word		
22				5th Word
23		7th Word		
24				6th Word
25		8th Word		
26				7th Word
27		9th Word		
28				8th Word
29		10th Word		
30				9th Word
31	Operand Fetch			
32	Execution			10th Word
33	Operand Write			

QUESTION 2:
A computer system has **64 KxWords** of main memory and a cache memory that can hold **4 KxWords** of data. Data transfers between main and cache memories are performed using blocks of **16 words**. The cache control unit uses the *two-way set associative mapping* technique where each set contains **two frames**.
In necessary cases, **FIFO** is used as the replacement algorithm.
Cache memory is used only for data, not for instructions.
The following questions (a and b) can be answered independently.

a) For write operations, the **Write Through** (WT) with **NO Write Allocate** (NWA) method is used.
Assume that the cache memory is empty at the beginning. The CPU runs the piece of pseudo code given on the right. Ten elements of array A are copied to array B. Each element is one word. The starting addresses of the arrays are given below:

```
For i = 0 to 9
    B[i] = A[i];
End of For
```

A: $083C
B: $0030

i) Which set/frames of cache memory does the cache control unit place arrays A and B into? Write your answers as decimal numbers. (Example: "**set number: 73, first frame**" or "**set number 85, second frame**.")

A: set number: 3, first frame and set number: 4, first frame
Note: Because of the NO Write Allocate (NWA) method, array B is not loaded to the cache.

ii) How many read misses, read hits, write misses, write hits, and block transfers occur during the run of the given loop? Write your answers in the boxes below:

Number of Read misses:	2	Number of Read hits:	8
Number of Write misses:	10	Number of Write hits:	0
Number of Block transfers:	2		

b) For write operations, the **Flagged Write Back** (FWB) with **Write Allocate** (WA) method is used.
Assume that the cache memory is empty at the beginning. The CPU runs the piece of pseudo code given on the right. In the first loop, ten elements of array X are copied to array Y. In the second loop, ten elements of array X are copied to array Z. Each element is one word. The starting addresses of the arrays are given below:

```
LOOP1:
For i = 0 to 9
    Y[i] = X[i];
End of For
LOOP2:
For i = 0 to 9
    Z[i] = X[i];
End of For
```

X: $0012
Y: $B010
Z: $0082

How many read misses, read hits, write misses, write hits, and block transfers occur **only** during the run of the **2nd loop** (LOOP2)? Do not count events in the first loop. Write your answers in the boxes below:

Number of Read misses in the **2nd loop**:	0	Number of Read hits in the **2nd loop**:	10
Number of Write misses in the **2nd loop**:	1	Number of Write hits in the **2nd loop**:	9
Number of Block transfers in the **2nd loop**:	1		

QUESTION 3:
In a symmetric multiprocessor (SMP) system with a shared bus, there are two CPUs (CPU1 and CPU2) that have local cache memories. The system does not include a shared L2 cache.
For write operations, the **Write Back** (WB) with **Write Allocate** (WA) method is used.
There is a shared variable A in the system. To provide cache coherence, the snoopy **MESI** protocol is used.

The following questions can be answered independently.

a) Valid copies of A reside in the cache of CPU1 and in main memory (A=1). The cache memory of CPU2 is currently empty.
What is the current MESI state of the corresponding frame in the cache of CPU1?

Exclusive

b) Valid copies of A reside in the cache of CPU1 and in main memory (A=1). The cache memory of CPU2 is currently empty. CPU2 reads the variable A.
What are the states of the corresponding frames in cache memories of CPU1 and CPU2 after the read operation?

CPU1: Shared CPU2: Shared

c) Valid copies of A reside in the cache of CPU1 and in main memory (A=1). The cache memory of CPU2 is currently empty. CPU2 writes to the variable A (A=2).
What are the states of the corresponding frames in cache memories of CPU1 and CPU2 after the write operation?

CPU1: Invalid CPU2: Modified