Yasin A. Yokus, Student No: 150190739

Question 1:

TIME(ns)	CPU	DMAC1	DMAC2	DMAC3	DMAC4
0	IF start				
20		DMA REQ	DMA REQ	DMA REQ	DMA REQ
40	IF end OR start	Data Transfer 1st Word START			
60	OR end				
140	EX start	DT 1st Word END	DT 1st-5th words Start		
180	EX end				
640		DT 2nd Word START	DT 1st-5th words End		
740		DT 2nd Word END		DT 1st-5th words Start	
1240		DT 3rd Word START		DT 1st-5th words End	
1340		DT 3rd Word END			DT 1st Word START
1440		DT 4th Word START			DT 1st Word END
1540		DT 4th Word END			DT 2nd Word START
1640		DT 5th Word START			DT 2nd Word END
1740		DT 5th Word END			DT 3rd Word START
1840	OW start				DT 3rd Word END
1870	OW end				DT 4th Word START
1970	IF start				DT 4th Word END
2010	IF end - EX start				DT 5th Word START
2110					DT 5th Word END

a) [1340]. Transfer of the 3rd word is completed at clock=1340ns because DMAC1 works on cycle-stealing mode, when the CPU gives access to DMAC1 it transfers only one word and returns control of bus back to the CPU. Since there are DMA requests from DMAC2 and DMAC3, CPU grant access to DMAC2 and DMAC1 and DMAC3. This adds extra waiting time for DMAC1 because DMAC2 and DMAC3 works on burst mode. Also, these DMACs are of flow-through type, they need 2 clock cycles. 100x5x2 for DMAC 2 and 3 results in 1000nanoseconds, 100x3 for 3 words of DMAC1 makes it 1300 ns, finally we add IF which is 40 ns. Thus 1340 nanoseconds is the answer. Because EX and OR cycles do not need memory access, they work in parallel with the DMACs.

b) [1870]. After IF cycle CPU gives access to DMAC 1. CPU continues to work on OR and EX cycles because they do not depend on memory. However, CPU waits DMACs to give access back because CPU needs memory access in order to execute OW cycle. OW cycle depends on access to the memory. After

DMACs transfer data both in burst mode in DMAC 2 and 3 and cycle stealing in DMAC 1 and 4, CPU gets access to the bus and OW cycle is executed. DMAC 4 can access the bus only after the end of OW cycle.

c) [2110]. In addition to explained steps above, DMAC 4 has lowest priority level. Thus, it waits for DMA requests of 1, 2 and 3 to end or executed once. Once there is not DMAC 2 and DMAC 3 blocking the access, DMAC 4 request can finally be met. However, DMAC 1 must execute Data Transfer once in order to give access to DMAC 4. That is why DMAC 4 waits until 1340 ns where DMAC1 finishes 3^{rd} word transfer. After this DMAC 1 and DMAC 4 transfers access to the bus between them. When the DMAC1 finishes transferring all 5 words, CPU gets access for the bus in order to execute OW cycle. Then DMAC4, IF, DMAC4 and EX together… At 2110 ns, DMAC4 completes transfer of all 5 words.

Question 2:

TABLE FOR QUESTION 2					
TIME(ns)	**CPU**	**DMAC1**	**DMAC2**	**DMAC3**	**DMAC4**
0	IF start				
10	Interrupt request				
20		DMA REQ	DMA REQ	DMA REQ	DMA REQ
40	IF end - OR start	Data Transfer 1st Word START			
60	OR end				
90	EX start	DT 1st Word End	DT 1st ...5th words START		
130	EX end				
340		DT 2nd Word Start	DT 1st ...5th words END		
390		DT 2nd Word End		DT 1st ...5th words START	
640		DT 3rd Word Start		DT 1st ...5th words END	
690		DT 3rd Word END			DT 1st ...5th words START
940		DT 4th Word Start			DT 1st ...5th words END
990	OW Start	DT 4th Word END	At the end of the instruction(end of OW cycle) CPU considers the interrupt but DMAC gets the bus, CPU can not start housekeeping operations which need memory. (From Lecture on May 6, 2021)		
1020	OW End	DT 5th Word Start			
1070	IR HOUSEKEEPING START	DT 5th Word END			
1270	IR HOUSEKEEPING END ------- ISR START				

a) [1070]. Since only DMAC which works on cycle-stealing mode is DMAC1, once DMAC2, 3 and 4 get access to the bus, we wait for 5x50 = 250 ns -> 3x250ns = 750 ns. DMAC1 can transfer 1 word between burst mode DMACs accesses. Finally when all the burst mode DMACs finishes transferring all 5 words, DMAC1 transfers its 4^{th} word. Since there is no DMAC other than DMAC1, bus access goes back to the CPU, because CPU needs bus access for execution of OW cycle. After OW cycle ends CPU considers the interrupt but DMAC gets access to the bus, CPU can not start housekeeping operations which need memory(register push, status register etc.).

b) [1270]. Housekeeping operations start at 1070 after DMAC1 finishes transferring 5th word. Since there is no DMA requests, Housekeeping operations take 200ns and ISR can start at 1270 ns. If we were to have DMA requests during Housekeeping operations, CPU could grant bus access to the DMAC when Housekeeping operations did not need the bus and we could say that ISR starts at 1270 earliest. But this is not the case for this question since there is no DMA request.