Ece Çınar
150150138

BLG 322E – Computer Architecture

Assignment 4

1) Truth Tables

a. BR output of bus arbiter

BR_1	BR_2	BR_3	BR_4	$BR_{arbiter}$
0	X	X	X	0
X	0	X	X	0
X	X	0	X	0
X	X	X	0	0
1	1	1	1	1

b. BG outputs of bus arbiter

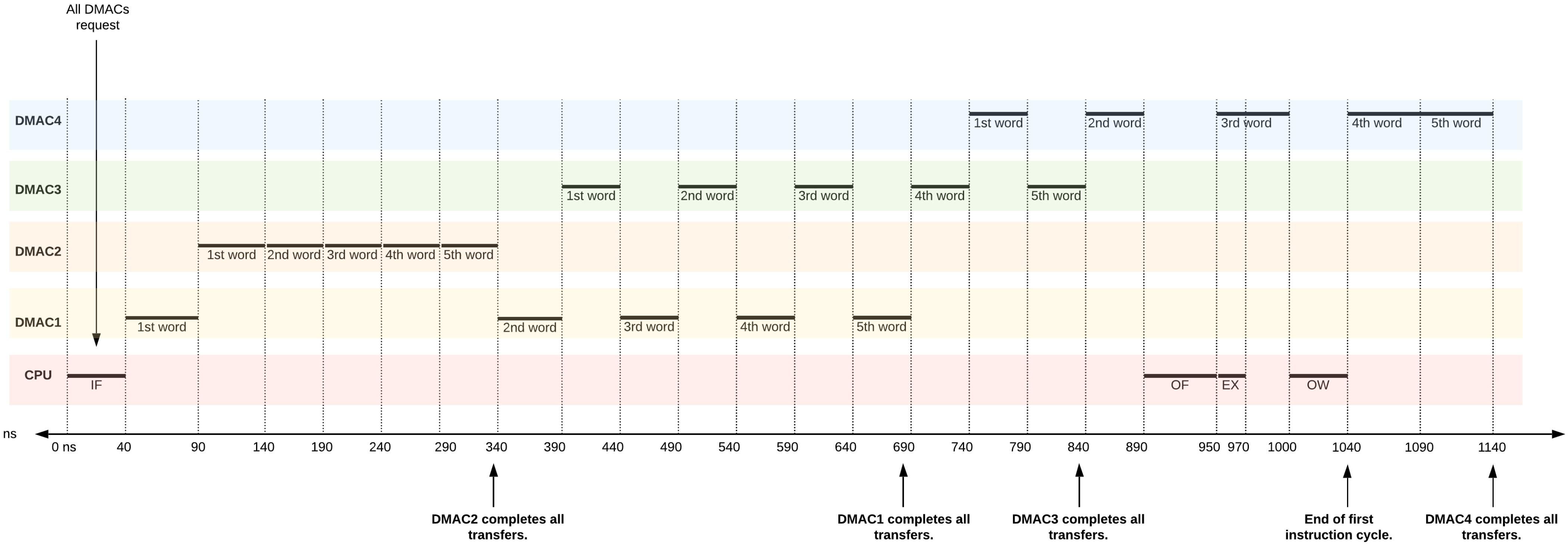
$BG_{arbiter}$	BR_1	BR_2	BR_3	BR_4	BG_1	BG_2	BG_3	BG_4
1	X	X	X	X	1	1	1	1
0	0	X	X	X	0	1	1	1
0	1	0	X	X	1	0	1	1
0	1	1	0	X	1	1	0	1
0	1	1	1	0	1	1	1	0
0	1	1	1	1	1	1	1	1

2) Logical Expressions

- $BR_{arbiter} = BR_1\ .BR_2\ .BR_3\ .BR_4$
- $BG_1 = BG_{arbiter} + BR_1$
- $BG_2 = BG_{arbiter} + BR_1\,' + BR_2$
- $BG_3 = BG_{arbiter} + BR_1\,' + BR_2\,' + BR_3$
- $BG_4 = BG_{arbiter} + BR_1\,' + BR_2\,' + BR_3\,' + BR_4$

.

3) DMAC and CPU Operations



All DMACs request

DMAC2 completes all transfers.

DMAC1 completes all transfers.

DMAC3 completes all transfers.

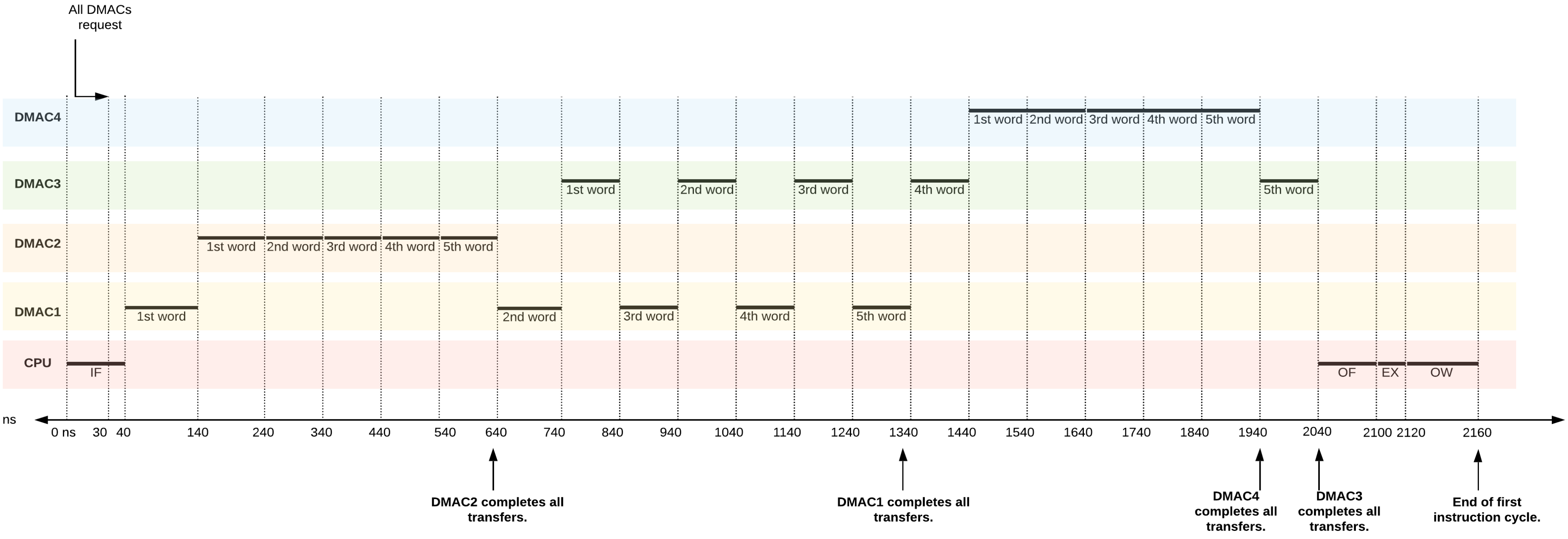
End of first instruction cycle.

DMAC4 completes all transfers.

Steps

- CPU completes **Instruction Fetch** (IF) cycle and releases the bus. (0ns – 40ns)
- DMAC$_1$ gets the bus and transfers its first word. It releases the bus because of its **cycle-stealing** mode. (40 ns – 90 ns)
- DMAC$_2$ gets the bus and transfers all five words one after another since it works in **burst** mode. (90 ns – 340 ns)
- DMAC$_1$ gets the bus and transfers its second word. It releases the bus because of its **cycle-stealing** mode. (340 ns – 390 ns)
- DMAC$_3$ gets the bus and transfers its first word. It releases the bus because of its **cycle-stealing** mode. (390 ns - 440 ns)
- DMAC$_1$ gets the bus and transfers its third word. It releases the bus because of its **cycle-stealing** mode. (440 ns – 490 ns)
- DMAC$_3$ gets the bus and transfers its second word. It releases the bus because of its **cycle-stealing** mode. (490 ns – 540 ns)
- DMAC$_1$ gets the bus and transfers its fourth word. It releases the bus because of its **cycle-stealing** mode. (540 ns – 590 ns)
- DMAC$_3$ gets the bus and transfers its third word. It releases the bus because of its **cycle-stealing** mode. (590 ns – 640 ns)
- DMAC$_1$ gets the bus and transfers its fifth word. This is the last word transfer of DMAC$_{1.}$ It releases the bus because word transfer operations are done. (640 ns – 690 ns)
- DMAC$_3$ gets the bus and transfers its fourth word. It releases the bus because of its **cycle-stealing** mode. (690 ns – 740 ns)
- DMAC$_4$ gets the bus and transfers its first word. It releases the bus because of its **cycle-stealing** mode. (740 ns – 790 ns)
- DMAC$_3$ gets the bus and transfers its fifth word. This is the last word transfer of DMAC$_{3.}$ It releases the bus because word transfer operations are done. (790 ns – 840 ns)
- DMAC$_4$ gets the bus and transfers its second word. It releases the bus because of its **cycle-stealing** mode. (840 ns – 890 ns)
- CPU gets the bus and completes **Operand Fetch** (OF) cycle. It releases the bus. (890 ns – 950 ns)
- DMAC$_4$ gets the bus and transfers its third word. Since **Execute** (EX) cycle doesn't require memory access, CPU handles this internal EX cycle at the same time with DMAC$_4$'s third word transmission. DMAC$_4$ releases the bus because of its **cycle-stealing** mode. (EX: 950 ns – 970 ns, DMAC$_4$: 950 ns – 1000 ns)
- CPU gets the bus and completes **Operand Write** (OW) cycle. OW is the last cycle of instruction cycles. **First instruction ends** at clock = 1040ns. (1000 ns – 1040 ns)
- After the instruction cycles end, DMAC$_4$ gets the bus and transfers its last two words. (1040 ns – 1140 ns)

4) DMAC and CPU Operations



i) DMAC$_2$ will complete transfer of the second word at clock = 340 ns. The operations performed until DMAC$_2$ transfers the second word are the same as third question. The only difference is DMAC's are **flow-through**. In flow-through type, data passes through the DMACs. DMACs have to access both I/O interface and the memory in order to transfer a word and that takes 50 ns + 50 ns = 100 ns time. So, the time DMAC$_2$ completes transfer of its second word can be calculated as below:

- clock = 40 ns (IF) + 100 ns (DMAC$_1$, one word) + 2 * 100ns (DMAC$_2$, two words) = 340 ns

ii) CPU will complete the first instruction cycle at clock = 2160 ns. In question three, DMAC$_4$ was working in cycle-stealing mode but here it is working in **burst** mode. Once it gets the bus, it transfers all five words it has and then grants the bus to DMAC$_3$ to transfer its last word. CPU is obliged to finish the OF, EX and OW cycles **after all words are transferred.**

- clock = 4 (all DMACs) * 5 (all words) * (100 ns) + (40 ns + 60 ns + 20 ns + 40 ns) (one instruction) = 2160 ns