a)

BR_1	BR_2	BR_3	BG	BR	BG_1	BG_2	BG_3
1	x	x	1	1	1	0	0
1	x	x	0	1	0	0	0
0	1	x	1	1	0	1	0
0	1	x	0	1	0	0	0
0	0	1	1	1	0	0	1
0	0	1	0	1	0	0	0
0	0	0	0	0	0	0	0

b)

$BG_1 = BR_1 BG$

$BG_2 = BR_1' BR_2 BG$

$BG_3 = BR_1' BR_2 'BR_3 BG$

$BR = BR_1 + BR_2 + BR_3$

c)

*$DMAC_1$, $DMAC_2$, $DMAC_3$ issues the request to the bus arbiter via BR

*Bus arbiter sends the request to 68000

*Bus arbiter gets BG from 68000 and sends signal to the $DMAC_1$

*After receiving BG, $DMAC_1$ observes AS signal and waits until previous bus cycle is complete

*If bus is free $DMAC_1$ asserts BGACK to inform the processor and the other DMACs about its mastership

*After being the master, $DMAC_1$ inactivates its BR_1 output to allow other DMACs sending their requests

*BG becomes 0

*Since $DMAC_2$ and $DMAC_3$ still sending BR request BG again becomes 1

*Bus arbiter gets BG from 68000 and sends it to the $DMAC_2$

*After receiving BG, $DMAC_2$ observes AS and BGACK signals and waits until the previous bus cyle is complete and the previous bus master releases the bus

*After transferring 1 word, $DMAC_1$ inactivates BGACK signal and again makes $BR_1 = 1$

* $DMAC_2$ asserts BGACK to inform others about its mastership

*After being the master, $DMAC_2$ inactivates its BR_2 output to allow other DMACs sending their requests

*BG becomes 0

*Since $DMAC_1$ and $DMAC_3$ still sending BR request BG again becomes 1

*Bus arbiter gets BG from 68000 and sends it to the $DMAC_1$

*After receiving BG, $DMAC_1$ observes AS and BGACK signals and waits until the previous bus cyle is complete and the previous bus master releases the bus

* $DMAC_2$ transfers 10 words and inactivates BGACK signal

* $DMAC_1$ asserts BGACK to inform others about its mastership

*After being the master, $DMAC_1$ inactivates its BR_1 output to allow other DMACs sending their requests

*BG becomes 0

*Since $DMAC_3$ still sending BR request BG again becomes 1

*Bus arbiter gets BG from 68000 and sends it to the $DMAC_3$

*After receiving BG, $DMAC_3$ observes AS and BGACK signals and waits until the previous bus cyle is complete and the previous bus master releases the bus

* $DMAC_1$ transfers 1 word and inactivates BGACK signal and makes $BR_1 = 1$;

*After that point $DMAC_1$ and $DMAC_3$ controls bus respectively until $DMAC_1$ completes transferring 10 words. At that point $DMAC_3$ has 2 more words to transfer and has the control of the bus. It makes $BR_3 = 0$, BG becomes 0, here there is no more BR request so 68000 can continue it's internal operations that does not require memory access but it is currently in fetch cycle which requires access to memory. So it continues waiting.

* Now $DMAC_3$ transfers 1 word and inactivates BGACK signal and makes $BR_3 = 1$

* Since $BR_3 = 1$ again BR becomes 1 and BG signal sent to $DMAC_3$ via bus arbiter.

* $DMAC_3$ asserts BGACK to inform the processor about its mastership

* After being the master, $DMAC_3$ inactivates its BR_3

* BG becomes 0

* Since there are no more BR request BG stays as 0

* $DMAC_3$ transfers its last word and now 68000 can continue its instruction since there are no more interrupt requests.