**a)**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| BR1 | BR2 | BR3 | BG | BR | BG1 | BG2 | BG3 |
| 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)**

BG1 = BR1 BG

BG2 = BR1’ BR2 BG

BG3 = BR1’ BR2 ‘BR3 BG

BR = BR1 + BR2 + BR3

**c)**

**\***DMAC1,DMAC2, DMAC3 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 DMAC1

**\***After receiving BG, DMAC1 observes AS signal and waits until previous bus cycle is complete

**\***If bus is free DMAC1 asserts BGACK to inform the processor and the other DMACs about its mastership

**\***After being the master, DMAC1 inactivates its BR1 output to allow other DMACs sending their requests

**\***BG becomes 0

**\***Since DMAC2 and DMAC3 still sending BR request BG again becomes 1

**\***Bus arbiter gets BG from 68000 and sends it to the DMAC2

**\***After receiving BG, DMAC2 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, DMAC1 inactivates BGACK signal and again makes BR1 = 1

**\*** DMAC2 asserts BGACK to inform others about its mastership

**\***After being the master, DMAC2 inactivates its BR2 output to allow other DMACs sending their requests

**\***BG becomes 0

**\***Since DMAC1 and DMAC3 still sending BR request BG again becomes 1

**\***Bus arbiter gets BG from 68000 and sends it to the DMAC1

**\***After receiving BG, DMAC1 observes AS and BGACK signals and waits until the previous bus cyle is complete and the previous bus master releases the bus

**\*** DMAC2 transfers 10 words and inactivates BGACK signal

**\*** DMAC1 asserts BGACK to inform others about its mastership

**\***After being the master, DMAC1 inactivates its BR1 output to allow other DMACs sending their requests

**\***BG becomes 0

**\***Since DMAC3 still sending BR request BG again becomes 1

**\***Bus arbiter gets BG from 68000 and sends it to the DMAC3

**\***After receiving BG, DMAC3 observes AS and BGACK signals and waits until the previous bus cyle is complete and the previous bus master releases the bus

**\*** DMAC1 transfers 1 word and inactivates BGACK signal and makes BR1 = 1;

**\***After that point DMAC1 and DMAC3 controls bus respectively until DMAC1 completes transferring 10 words. At that point DMAC3 has 2 more words to transfer and has the control of the bus. It makes BR3 = 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 DMAC3 transfers 1 word and inactivates BGACK signal and makes BR3 = 1

**\*** Since BR3 = 1 again BR becomes 1 and BG signal sent to DMAC3 via bus arbiter.

**\*** DMAC3 asserts BGACK to inform the processor about its mastership

**\*** After being the master, DMAC3 inactivates its BR3

**\*** BG becomes 0

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

**\*** DMAC3 transfers its last word and now 68000 can continue its instruction since there are no more interrupt requests.