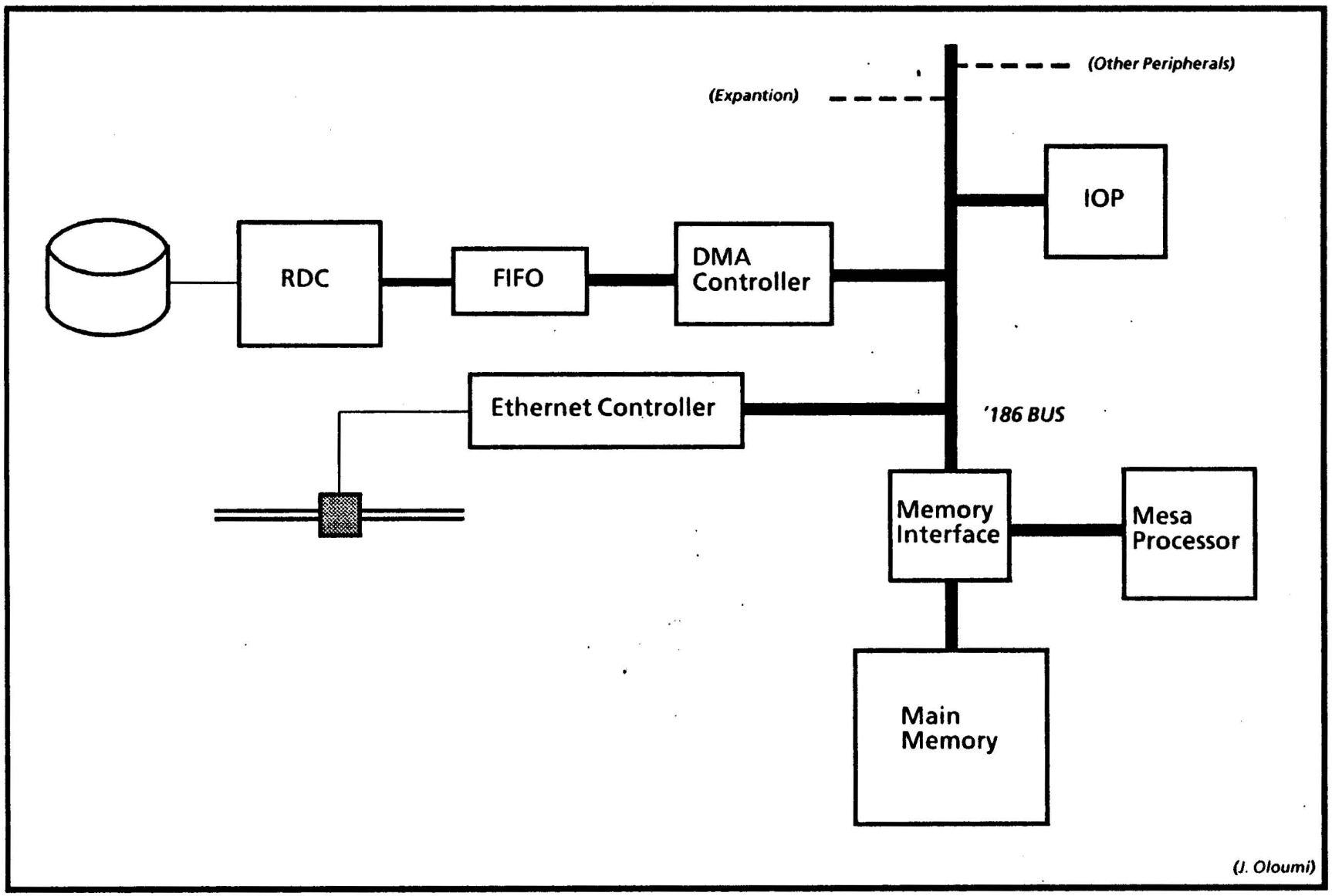

Dove Block Diagram

DMA Features:

- **Programmable by IOP (Starting Address, Word Count, CR Register and Start DMA)**
- **Can transfer up to 256 words (of 16 bits) at a time**
- **Provides 24 bits of Address to the Main Memory**
- **Data transfer at '186 memory cycle rate**
- **End-of-transfer always interrupts IOP**

- ***What is being "DMAed"?***
  - Pages of data (256 words/page = mesa page)
  - IOCBs

- ***Limitations of DMA***

Treats everything as DATA to be DMAed between FIFO and Main Memory (process data and IOCB).

- ***Programming the DMA***
  - -Word Count
  - Statrting Address (24 bits)
  - Transfer Direction
  - StartDMA
  - Other (2942, etc.)
  - Have the Programmer's Notes handy

- ***Control and Status Registers***
  - Reset-on-read Status Register
  - Control Register

- ***Interfacing '186 bus:*** (Playing '186 role for Main Memory, S' Lines)
  - DMA Controller is a Bus Master
  - DMA Controller acts as an 80186 for the Main Memory during DMA transfer,
  - And acts as an IOP peripheral other times (available for programming and/or inquiries).
  - Bus Arbiter under program control grants or removes Bus Masters according to a priority scheme. Arbiter signals DMA Controller to leave '186 bus by dropping DMA's Hold Acknowledge line (HLDA).

- ***DMA's response to Special situations:***
  - FIFO not responding (out of bound condition based on direction) :
    Drops HLD and waits for FIFO available before raises HLD again.
  - Ethernet intervention:
    Drops HLD and waits for one more T cycle before raising HLD (Bus Arbiter will delay granting the bus until the higher priority user relinquishes the bus).

PromStatesRevP.DMA
Last edited by Jafil Oloumi 23 May 84 16:39:53
Notes:
1 DRead' & DWrite' have been combined into one signal and inverted. DWrite/DRead goes high beginning I2 and stays high until the end of I3
2 Based on the state of FIFOAtOfBound' during I3, decision is made to put the Status on the bus beginning I4 for the next cycle.
3 Another signal (FIFOPrefetch) added to pre fetch the FIFO during FIFO to MEM transfer.

Comments	Unused bit	Disk HLDA	Sync ARDY	FIFO O.B.'	EndOf Xfer'	State Number		Error Trap'	Int Req	Gate SFlines'	ADDr'/ Data	DMA Active	DWrite DRead	Disk HLD	FIFO Pr'fetch	Unused Data	Next State
InitialWait	X	X	X	X	X	0 0000	↑	1	0	1	1	0	0	0	0	000	0 1100
SendHLD	X	X	X	X	1	0 0001	↑	1	0	1	1	0	0	1	0	000	0 0010
Error1	X	X	X	X	0	0 0001	↑	0	1	1	1	0	0	0	0	000	0 1101
WaitForHLDA	X	0	X	X	X	0 0010	↑	1	0	1	1	0	0	1	0	000	0 0010
GotHLDA	X	1	X	X	X	0 0010	↑	1	0	1	1	1	0	1	0	000	0 0011
IsState	X	X	X	X	X	0 0011	↑	1	0	0	1	1	0	1	1	000	0 0100
I1State	X	X	X	X	X	0 0100	↑	1	0	0	0	1	1	1	1	000	0 0101
I2State	X	X	X	X	X	0 0101	↑	1	0	0	1	1	1	1	0	000	0 0110
I3	X	X	X	X	X	0 0110	↑	1	0	1	1	1	1	1	0	000	0 0111
IwAndWait	X	X	0	X	X	0 0111	↑	1	0	1	1	1	1	1	0	000	0 0111
I4Normal	X	1	1	1	1	0 0111	↑	1	0	0	1	1	0	1	1	000	0 0100
I4AndEOXfer	X	X	1	X	0	0 0111	↑	1	0	1	1	1	0	1	0	000	0 1010
I4AndOutOfBnd	X	X	1	0	1	0 0111	↑	1	0	1	1	1	0	1	0	000	0 1000
I4AndEtherHld	X	0	1	X	1	0 0111	↑	1	0	1	1	1	0	1	0	000	0 1000
DropHLD	X	0	X	X	X	0 1000	↑	1	0	1	1	1	0	0	0	000	0 1001
ItIsFIFO	X	1	X	X	X	0 1000	↑	1	0	1	1	1	0	0	0	000	0 1100
Delay	X	X	X	X	X	0 1001	↑	1	0	1	1	0	0	0	0	000	0 0001
SendIntReq	X	X	X	X	X	0 1010	↑	1	1	1	1	1	0	0	0	000	0 1011
EndDMA	X	X	X	X	X	0 1011	↑	1	0	1	1	0	0	0	0	000	0 0000
WaitForFIFO	X	X	X	0	X	0 1100	↑	1	0	1	1	0	0	0	0	000	0 1100
FIFOAvailable	X	X	X	1	X	0 1100	↑	1	0	1	1	0	0	0	0	000	0 0001
ErrorLoop	X	X	X	X	X	0 1101	↑	0	1	1	1	0	0	0	0	000	0 1101
OtherStates	X X X X X (State numbers "0 1101" to "1 1111")						↑	0	1	1	1	0	0	0	0	000	0 1101

A Transfer Scenario (Mem-FIFO)

IOP:

- Programs the DMA (not necessarily in this order):
  CR register, Word Count (2's complement), Starting Address (24 bits)
- Sets the direction bit to 1
- Issues AllowRDC
- Issues StartDMA

DMA:

- Checks if FIFO is available for transfer
- Sends Hold request (HLD) and waits for HLDA
- Once HLDA received, transfers one word per memory cycle:
  Moves S' lines active (memory Read)
  Puts 24 bits address on the bus (AD15-00 and AA23-16) during T1 state
  Waits till data from Memory stable on the bus (delays for ARDY)
  Data written into destination (FIFO) by T4 state
- Decrements the Word Count (increments the 2's complement)
- Increments the address (24 bits)
- Transfers another word (during the next T1 through T4)
- If HLDA drops, DMA drops HLD, delays for one cycle, sends Hold request again
- When FIFO unavailable, DMA drops HLD, waits for FIFO available, then sends HLD
- DMA continues to transfer until End-of-Transfer
- At End-of-Transfer, sends Interrupt to IOP and go back to ready state (InitialWait).

IOP:

- Once interrupted will check if it is End-of-Transfer or due to error.
- If IOP is interrupted while DMA, it should issue AllowRDC again (to Arbiter)
- IOP will have the control for the next DMA transfer.

A Transfer Scenario (FIFO-Mem)

IOP:

- Programs the DMA (not necessarily in this order):
  CR register, Word Count (2's complement), Starting Address (24 bits)
- Sets the direction bit to 0
- Issues AllowRDC
- Issues StartDMA

DMA:

- Checks if FIFO is available for transfer
- Sends Hold request (HLD) and waits for HLDA
- Once HLDA received, transfers one word per memory cycle:
  Moves S' lines active (memory Write)
  Puts 24 bits address on the bus (AD15-00 and AA23-16) during T1 state
  Pre-Fetches data from FIFO (during last T4 & present T1)
  Puts data on the '186 bus during T2 and T3 (Plus any Tw's caused by ARDY)
  Data written into destination (Main Memory) by T4 state
- Decrements the Word Count (increments the 2's complement)
- Increments the address (24 bits)
- Transfers another word (during the next T1 through T4)
- If HLDA drops, DMA then drops HLD, delays for one cycle, sends Hold request again
- When FIFO unavailable, DMA drops HLD, waits for FIFO available, then sends HLD
- DMA continues to transfer until End-of-Transfer
- At End-of-Transfer, sends Interrupt to IOP and go back to ready state (InitialWait).

IOP:

- Once interrupted will check if it is End-of-Transfer or due to error.
- If IOP is interrupted while DMA, it should issue AllowRDC again (to Arbiter)
- IOP will have the control for the next DMA transfer.

FIFO Features:

- **512 words long (made of 512X9 components)**
- **Bidirectional access (memory-to-disk & disk-to-memory)**
- **Programable direction control**
- **Simultaneously and asynchronously accessible by DMA and RDC**
- **Full, Empty and Half Full indication**