The Daisy memory will be constructed using 256K DRAM chips. These will have a "nibble mode" feature that allows one to read four bits from a chip in quick succession. This mode will be used by the display controller and by the processors when doing double word fetches. Unfortunately, the method of addressing the nibbles is clumsy. Assume the address bits are held in an RH,,R,,FBus triplet, one byte in each. The RH bits appear on the H bus and the R bits on the B bus.

H.[0..7]								B.[00..05]								F.[08..15]							
0	1	2	3	4	5	6	7	00	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23

The 256K memory chips take 18 bit addresses in two 9 bit groups, Row address and Column address. The remaining bits are either unused, used to select an A chip (AS.x), or used to select a memory bank(BS). Assume addressing were done in the straightforward manner:

unused		AS.[0..2]			BS	RA.[0..8]									CA.[0..8]								
0	1	2	3	4	5	0	1	2	3	4	5	6	7	8	0	1	2	3	4	5	6	7	8
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23

The unfortunate part is that the four bits returned from each memory chip are addressed by ColAddr.8,,RowAddr.8. Thus, if address 000000 were sent to a memory system addressed in this manner, words 0, 512, 1, 513 would be returned on successive nibble mode accesses. This is horribly confusing for either BitBlt, the display controller or both. To fix this, the address bits are permuted as follows:

0	1	2	3	4	5	6	7	8	9	10	11	12	13	23	14	15	16	17	18	19	20	21	22
0	1	2	3	4	5	6	7	00	01	02	03	04	05	15	06	07	08	09	10	11	12	13	14
H.[0..7]								B.[00..05]						F	B	F.[07..14]							
0	1	2	3	4	5	0	1	2	3	4	5	6	7	8	0	1	2	3	4	5	6	7	8
unused		AS.[0..2]			BS	RA.[0..8]									CA.[0..8]								

The A Chip, Bank Select and RA bits are needed first. For obscure reasons, they are permuted when sent from the IBIP to the A chip (see below). The chips accept the CA.[0.8] bits some 50 nS later. Thus, it is only necessary to sent 15 bits in one step and then 9 bits in the second step. Because we wish to be able to support 64K chips if necessary, we send 16 bits, then 8. The A chip saves the extra bit sent in the first step and sends it as part of the column address.

During a real memory reference, the most significant 8 bits are read from an RH register (H.[0..7]), the next most significant bits from an R register (B.[00..07]) and the least significant 8 bits from the F bus (F.[08..15]). Arithmetic or logical operations are only allowed on the least significant byte of real addresses. This in only reasonable since pages are 256 words long and consecutive virtual pages are seldom assigned consecutive real pages.

The AP interface pins are multiplexed among the first group of address bits (called A1), the second group of address bits (called A2) and the write or read data. The two address groups are formed as follows:

	AP.00	AP.01	AP.02	AP.03	AP.04	AP.05	AP.06	AP.07	AP.08	AP.09	AP.10	AP.11	AP.12	AP.13	AP.14	AP.15
A1	B.00	B.01	B.02	B.03	B.04	B.05	B.06	H.0	H.1	H.2	H.3	H.4	H.5	H.6	H.7	F.15
Mem. Addr.	RA.2	RA.3	RA.4	RA.5	RA.6	RA.7	CA.0	unused	unused	AS.0	AS.1	AS.2	BS	RA.0	RA.1	RA.8
A2								F.7*	F.8	F.9	F.10	F.11	F.12	F.13	F.14	
Mem. Addr.								CA.1	CA.2	CA.3	CA.4	CA.5	CA.6	CA.7	CA.8	

* F.7 = B.7 since the most significant byte of the ALU always performs F_rB when MAR_ is specified.

"Mem Addr" is the interpretation the A chip gives to the address bits on the AP bus. The presence of the A1 bits is signalled by Cp' going LO. A2 is signalled by Ca' dropping.

Continued from IBIPSim41.sily

The other source of addresses for the AP bus is the Instruction Buffer. Its addresses are generated by the concatenation of the fetch Program Counter's page number (fPCp) and its page displacement (fPCd). These bits are sent as follows:

	AP.00	AP.01	AP.02	AP.03	AP.04	AP.05	AP.06	AP.07	AP.08	AP.09	AP.10	AP.11	AP.12	AP.13	AP.14	AP.15
A1	fPCp.0	fPCp.1	fPCp.2	fPCp.3	fPCp.4	fPCp.5	fPCp.6	fPCp.8	fPCp.9	fPCp.10	fPCp.11	fPCp.12	fPCp.13	fPCp.14	fPCp.15	0
Mem.	RA.2	RA.3	RA.4	RA.5	RA.6	RA.7	CA.0	unused	unused	AS.0	AS.1	AS.2	BS	RA.0	RA.1	RA.8
A2								fPCp.7	fPCd.0	fPCd.1	fPCd.2	fPCd.3	fPCd.4	fPCd.5	fPCd.6	
Mem.								CA.1	CA.2	CA.3	CA.4	CA.5	CA.6	CA.7	CA.8	

Note that since all Instruction Buffer fetches reference double words, fPCd.7 is always sent as 0 or GND.

Map _

The Mesa Virtual Memory map is held in main memory starting at location 1000'X. The A chip has no special circuitry for addressing the map, virtual addresses are converted into map indexes in the P chip. The 80186 must do a similar transformation. The entries for consecutive virtual pages do not lie in consecutive map locations. This again result from the permutation of address bits done to make nibble mode accesses return consecutive addresses. The trick here was to send the least significant bit of address in the first group of addresses (A1). The least significant bit of a map index would normally be F.7 since this is the least significant bit of a computed page number. Unlike F.15 however, F.7 will not be ready into well into Phase B. This is too late for it to be sent in A1. The only bits that will be ready in Phase A are the RH.[0..7] bits. Conceptually, any of them could be chosen to be the least significant map index bit. To make things easy for the 80186, we should choose either RH.0 or RH.7. To convert a virtual page number into a map index, the 80186 will move the bit chosen (RH.0 or RH.7) to the least significant bit and shift the lower order bits up to fill the hole. One would like to choose RH.0 since this becomes a simple 16 bit left rotate. However, if a 22 or 23 bit virtual address space is used, only even locations in the map will be used, wasting 16K or 32K words respectively. Hence, we choose RH.7. The 80186 can form the map index by doing a right rotate of the upper byte, putting the bit to be moved into the Carry flag followed by a 16 bit left rotate, putting the Carry flag into the LSB.

	AP.00	AP.01	AP.02	AP.03	AP.04	AP.05	AP.06	AP.07	AP.08	AP.09	AP.10	AP.11	AP.12	AP.13	AP.14	AP.15
A1	H.0	H.1	H.2	H.3	H.4	H.5	H.6	0	0	0	0	0	0	0	1	H.7
Mem.	RA.2	RA.3	RA.4	RA.5	RA.6	RA.7	CA.0	unused	unused	AS.0	AS.1	AS.2	BS	RA.0	RA.1	RA.8
A2								F.0	F.1	F.2	F.3	F.4	F.5	F.6	F.7	
Mem.								CA.1	CA.2	CA.3	CA.4	CA.5	CA.6	CA.7	CA.8	

"Mem" is the interpretation the A chip gives to the address bits on the AP bus. The presence of the A1 bits is signalled by Cp' going LO. A2 is signalled by Ca' dropping.