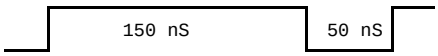
1. The MSI IBIP Simulator does not use the asynchronous state machines of the IBIP itself. This was done because of the difficulty of constructing such hazard-free logic from MSI. Instead, the MSI machine is entirely synchronous. There are two main clocks in the machine, the Emulator Clock, or EmuClk, and the Bus State machine clock or BusClk. The Emulator clock can be stopped on clock boundries. It has a 75% duty cycle:



2. Like the Dandelion, the EmuClk is stopped by deleting the LO pulse. Thus the decision about whether to stop the clock must be made before the LO pulse.

3. The BusClk is a 50 nS, 50% duty cycle clock. The Bus state machine attempts to mimic the P Chip state machine in that it goes through the same states in response to the same conditions. It contains one flip-flop element per state, so may more correctly be viewed as a token-passing machine.

4. The Bus State machine in the simulator may only change its outputs on the rising edge of the clocks. A Bus read cycle is shown below.

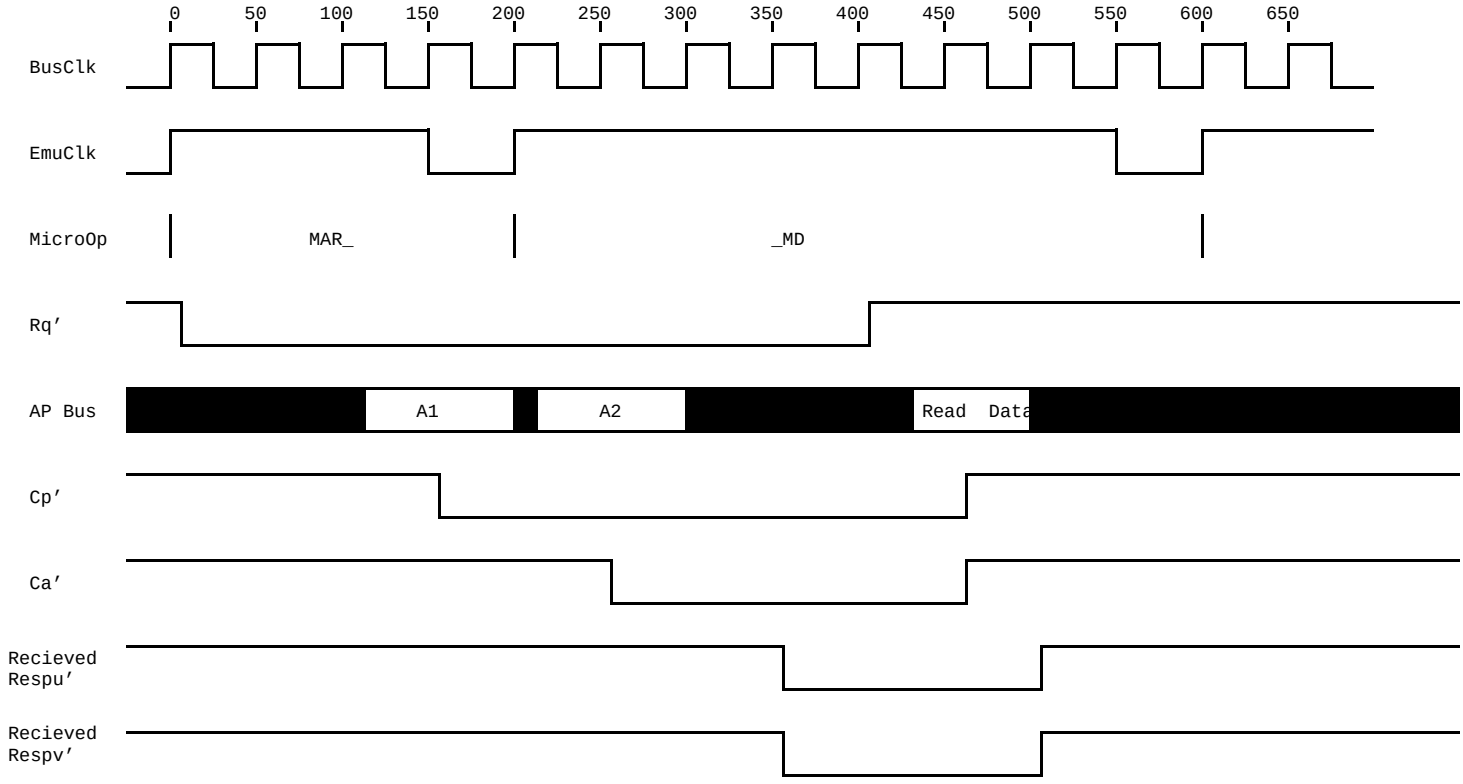
John Dillon's preliminary estimate of the Ca' to Resp' time is 195 nS (8/15/83).

If three 200 nS micronistruction are used to read memory, the memory read time is 600 nS. Assuming display contention adds ~123 nS to each access, an average memory read takes 723 nS or about 57% Dandelion speed.

Note that read data is actually on the AP Bus between 50 and 100 nS before the synchronized version of Respu' and Respv' reach the Bus State Machine. The data is latched in the Simulator's Connector Card. The exact duration of the read data on the AP bus is not known.

5. The sequence is:

0 nS	Next microinstruction contains a MAR_ (or Map_) clause, so Rq' goes LO and AP Bus output multiplexer sends A1 address for a MAR_.
50 nS	No Change
100 nS	No Change, Wait for F.15 to become valid.
150 nS	Send Cp' LO.
200 nS	Capture and send MAR_ type A2 address on AP Bus.
250 nS	Send Ca' LO.
300 nS	No Change
350 nS	Assuming Respu' and Respv' were sent within 50 nS of Ca' being sent, they will arrive through the synchronizers now.
400 nS	Set Rq' HI
450 nS	Set Ca', Cp' HI
500 nS	Assuming < 200 nS access from Ca' (including A Chip delays), the synchronized versions of Respu' and Respv' arrive back now. They latch the read data into Respv' (we must remember that this is a single fetch).
550 nS	Return to the idle state (S2). Release EmuClk if it was being held waiting for the memory reference.



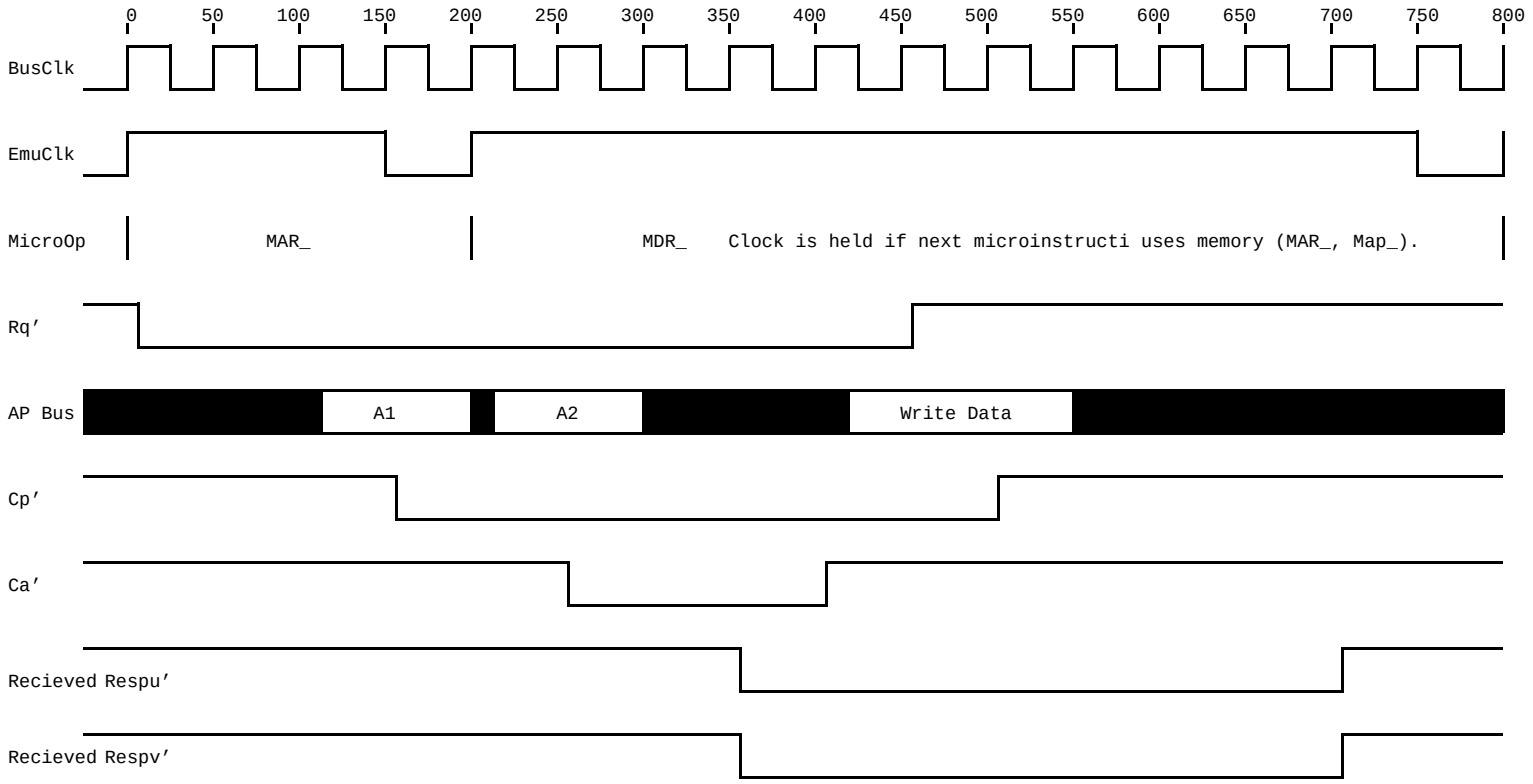
XEROX SDD	Project	Reference	File	Designer	Rev	Date	Page
	Daisy	Simulator - AP Bus Timing Single Fetch Operation	IBIPSim60.sily	Davies	A	8/18/83	60

6. A Memory Write operation is shown below. The sequence is:

0 nS	Next microinstruction contains a MAR_ (or Map_) clause, so Rq' goes LO and AP Bus output multiplexer sends A1 address for a MAR_.
50 nS	No Change
100 nS	No Change, Wait for F.15 to become valid.
150 nS	Send Cp' LO.
200 nS	Capture and send MAR_ type A2 address on AP Bus.
250 nS	Send Ca' LO.
300 nS	No Change
350 nS	Assuming Respu' and Respv' were sent within 50 nS of Ca' being sent, they will arrive through the synchronizers now.
400 nS	Ca' is driven HI. Capture the Write Data, it be ready now. Drive Write Data onto AP Bus.
450 nS	Rq' is Set HI, since Ca' is already HI, the A chip knows the operations is a write.
500 nS	Cp' is driven HI, signalling the write Data is available. We cannot do this at 450 nS as that could confuse the A Chip. If the A Chip thought Cp' went HI before Rq', it would think the current cycle was a aborted.
550 nS	The Write Data is removed from the AP Bus.
650 nS	Assuming a 130 nS delay in the A Chip from Cp' going HI to Respv' going HI, the first synchronizing FF sees Respv' now.
700 nS	The second FF in the synchronizing chain sends Respv' and Respu' now.
750 nS	Return to the idle state (S2). Release EmuClk if it was being held waiting for the memory reference.

A write operation is ~411/800 nS or 51% Dandelion speed assuming no display or I/O contention. If the display adds an average of 123 nS to every P Chip reference (SEE SR our average write cycle is 923 nS or 44% Dandelion speed.

The 130 nS delay from Cp' HI to Respv' HI is a preliminary estimate from John Dillon (8/15/83).



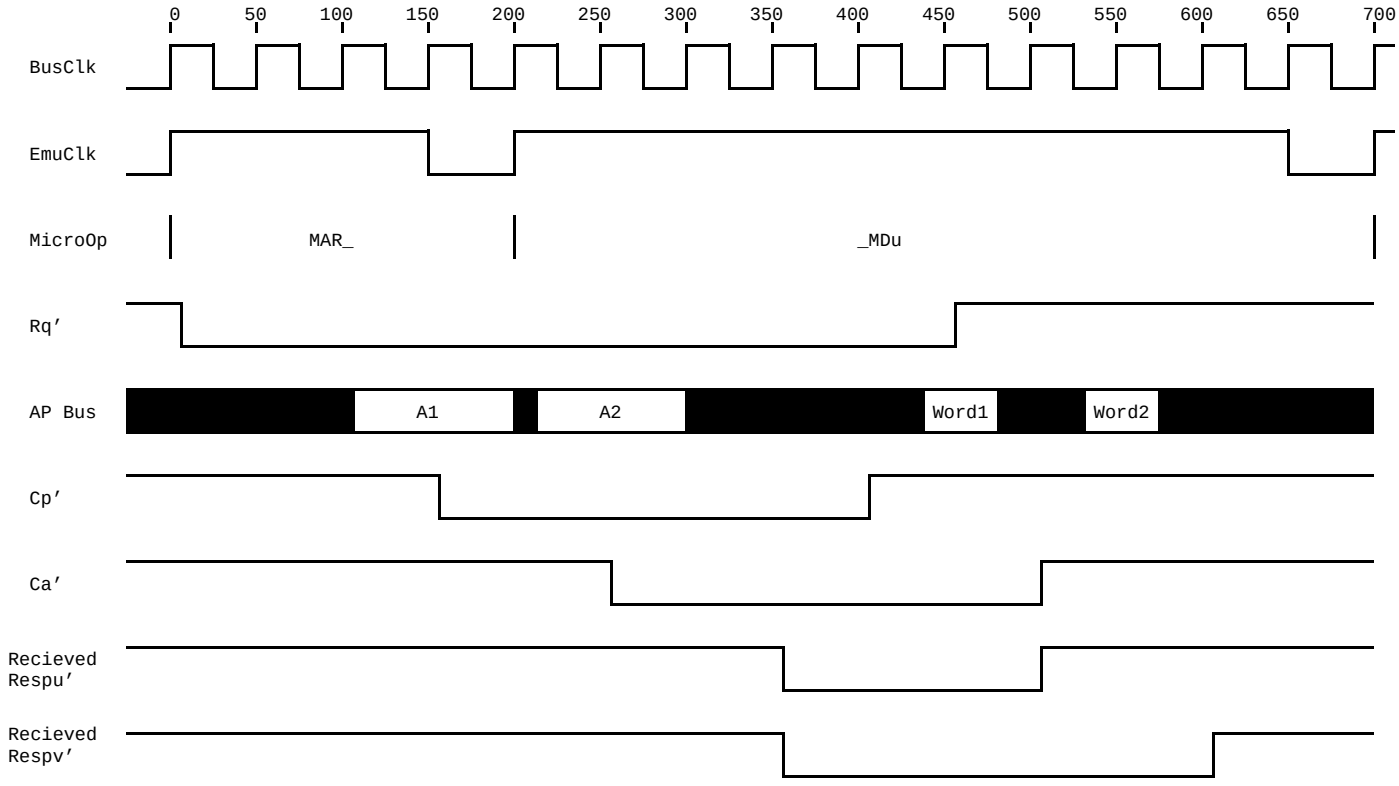
7. An Emulator-initiated double word fetch operation is shown below. The sequence is:

0 nS	Next microinstruction contains a MAR_ (or Map_) clause, so Rq' goes LO and AP Bus output multiplexer sends A1 address for a MAR_.
50 nS	No Change
100 nS	No Change, Wait for F.15 to become valid.
150 nS	Send Cp' LO.
200 nS	Capture and send MAR_ type A2 address on AP Bus.
250 nS	Send Ca' LO.
300 nS	No Change
350 nS	Assuming Respu' and Respv' were sent within 50 nS of Ca' being sent, they will arrive through the synchronizers now.
400 nS	Set Cp' HI.
450 nS	Set Rq' HI. Since Cp' is already HI, the A chip knows this is a double words fetch.
500 nS	Assuming < 200 nS access from Ca' (including A Chip delays), the synchronized version of Respu' arrives back now. It latches the first word of read data into the MDu. The State machine also raises Ca' at this time.
550 nS	No Change
600 nS	Assuming Respv' is raised within 150 nS of Respu', the buffered version of Respv' will be seen now.
650 nS	Return to the idle state (S2). Release EmuClk if it was being held waiting for the memory reference.

John Dillon's preliminary estimate of the Ca' to Resp' time is 195 nS (8/15/83).

His preliminary estimate of the Respu' HI to Respv' HI time is 140 nS (8/15/83).

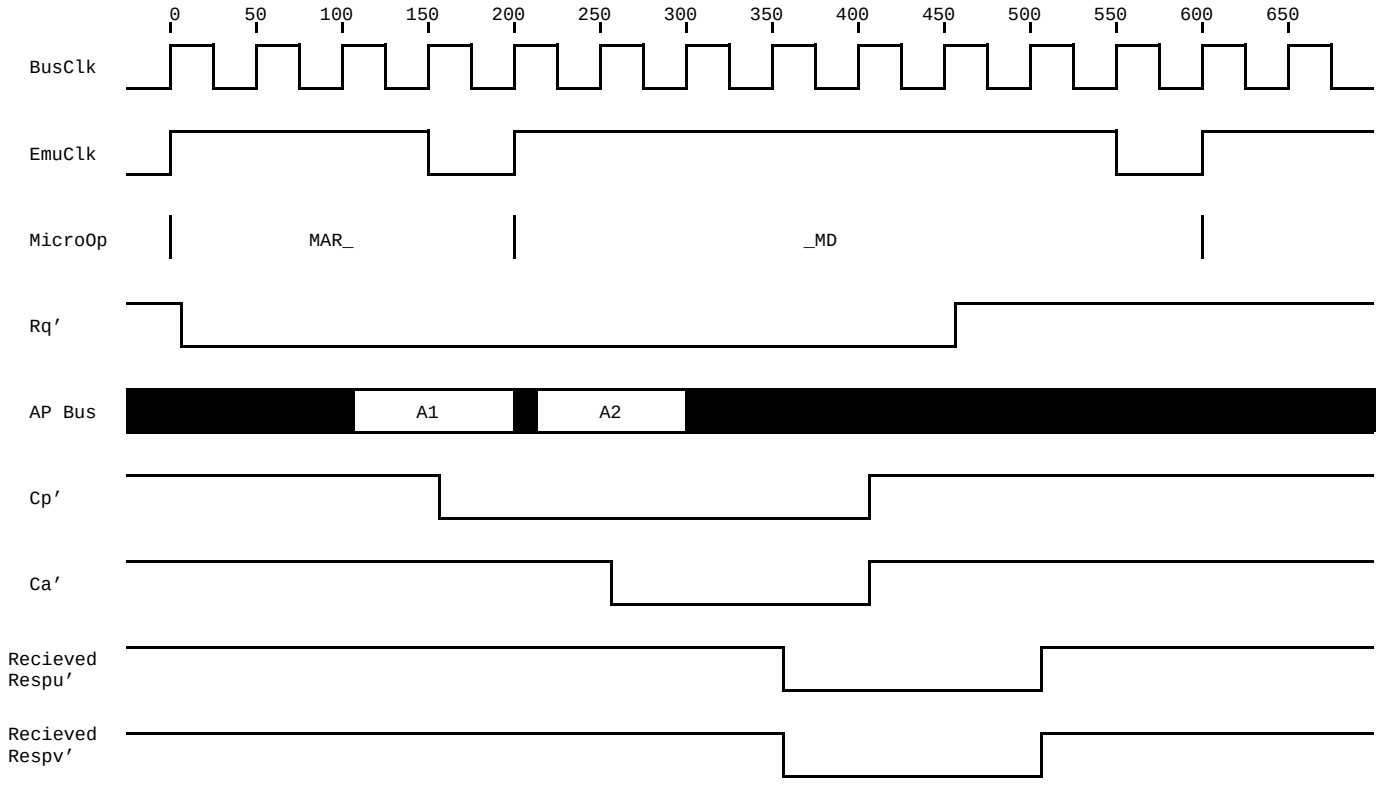
Note that read data is actually on the AP Bus between 50 and 100 nS before the synchronized version of Respu' and Respv' reach the Bus State Machine. The data is latched in the Simulator's Connector Card. The exact duration of read data on the AP Bus is not known.



8. An Emulator-initiated abort operation is shown below. The sequence is:

0 nS	Next microinstruction contains a MAR_ (or Map_) clause, so Rq' goes LO and AP Bus output multiplexer sends A1 address for a MAR_.
50 nS	No Change
100 nS	No Change, Wait for F.15 to become valid.
150 nS	Send Cp' LO.
200 nS	Capture and send MAR_ type A2 address on AP Bus.
250 nS	Send Ca' LO.
300 nS	No Change
350 nS	Assuming Respu' and Respv' were sent within 50 nS of Ca' being sent, they will arrive through the synchronizers now.
400 nS	Set Cp', Ca' HI.
450 nS	Set Rq' HI. Since Cp' and Ca' are already HI, the A Chip knows to abort this operation.
500 nS	Assuming < 200 nS access from Ca' (including A Chip delays), the synchronized version of Respu' and Respv' arrive back now ending the memory cycle.
550 nS	Return to the idle state (S2). Release EmuClk if it was being held waiting for the memory reference.

John Dillon's preliminary estimate of the Ca' to Resp' time for an Abort operation is ~170 nS (8/15/83).



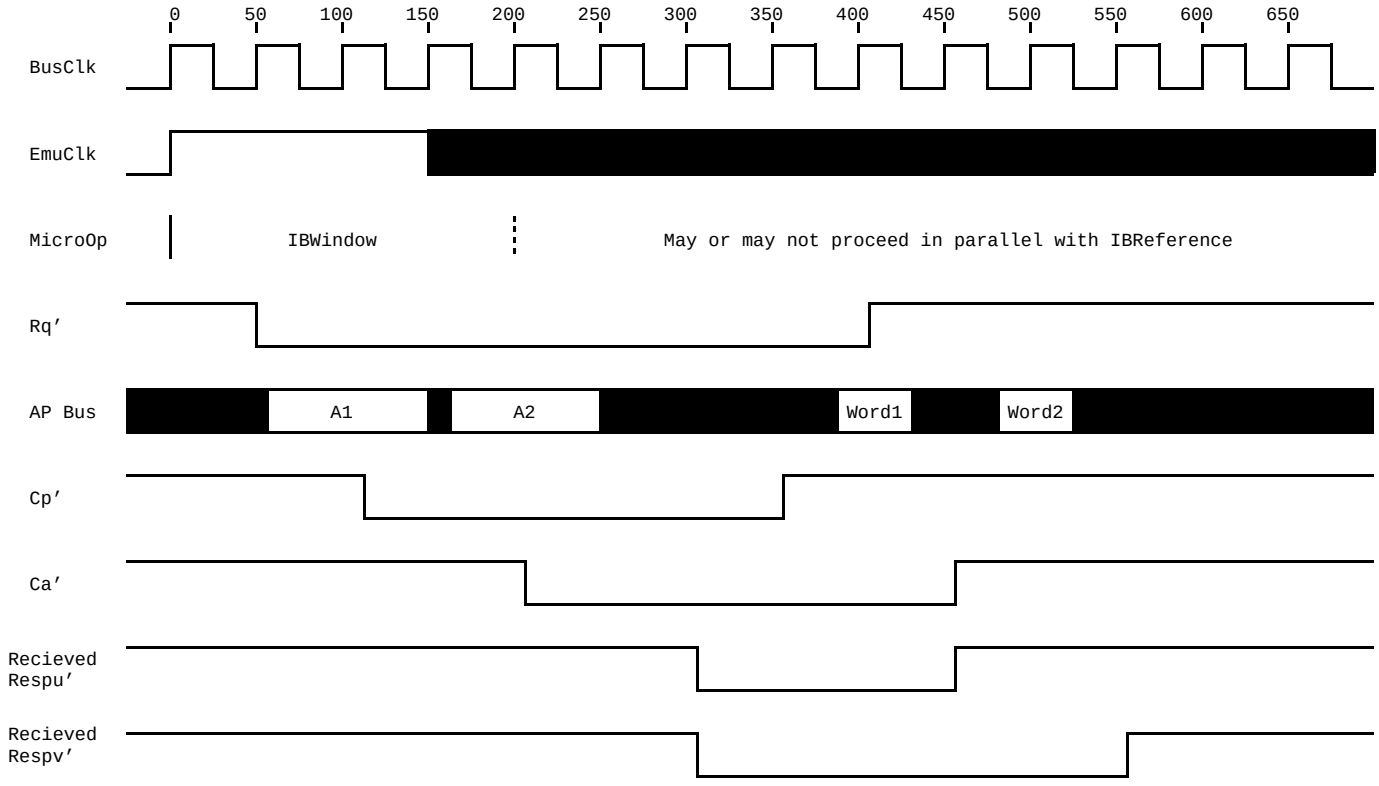
9. An IBReference caused by an IBWindow clause is shown below.

0 nS	Current microinstruction contains an IBWindow clause, there is room in the IB for another double word, fPCAtEndOfPage is false and the IB is enabled.
50 nS	Leave S2 and send Rq', gate fPCp on bus as A1.
100 nS	Send Cp' LO
150 nS	Send fPCd as A2 address on AP Bus. This is provided via a transparent latch.
200 nS	Disable A2 Latch, Send Ca'
250 nS	No Change
300 nS	Assuming Respu' and Respv' were sent within 50 nS of Ca' being sent, they will arrive through the synchronizers now.
350 nS	Set Cp' HI.
400 nS	Set Rq' HI. Since Cp' is already HI, the A chip knows this is a double words fetch.
450 nS	Assuming < 200 nS access from Ca' (including A Chip delays), the synchronized version of Respu' arrives back now. It latches the first word of read data into the even word of a double word in the IB. The State machine also raises Ca' at this time.
500 nS	Increment fPCd
550 nS	Assuming Respv' is raised within 150 nS of Respu', the buffered version of Respv' will be seen now. This stores the data into the odd word of the double IB word.
600 nS	Set the Full bit corresponding to the IB double word just filled.
650 nS	Return to the idle state (S2). Release EmuClk if it was being held waiting for the IBReference.

John Dillon's preliminary estimate of the Ca' to Resp' time is 195 nS (8/15/83).

His preliminary estimate of the Respu' HI to Respv' HI time is 140 nS (8/15/83).

Note that read data is actually on the AP Bus between 50 and 100 nS before the synchronized version of Respu' and Respv' reach the Bus State Machine. The data is latched in the Simulator's Connector Card. The exact duration of read data on the AP Bus is not known.



XEROX SDD	Project	Reference	File	Designer	Rev	Date	Page
	Daisy	Simulator - AP Bus Timing IBReference from IBWindow	IBIPSim64.sily	Davies	A	8/18/83	64

10. An IBReference caused by attempting to read an the IB when it is empty is shown below.

-150 nS	Current microinstruction starts executing.
0 nS	The next microinstruction contains some sort of IBAccess (_ib,ibNA, ... ,IBDisp, AwIBDisp), IBEmpty is true, fPCAtEndOfPage is false and the IB is enabled.
50 nS	Leave S2 and send Rq', gate fPCp on bus as A1.
100 nS	Send Cp' LO
150 nS	Send fPCd as A2 address on AP Bus. This is provided via a transparent latch.
200 nS	Disable A2 Latch, Send Ca'
250 nS	No Change
300 nS	Assuming Respu' and Respv' were sent within 50 nS of Ca' being sent, they will arrive through the synchronizers now.
350 nS	Set Cp' HI.
400 nS	Set Rq' HI. Since Cp' is already HI, the A chip knows this is a double words fetch.
450 nS	Assuming < 200 nS access from Ca' (including A Chip delays), the synchronized version of Respu' arrives back now. It latches the first word of read data into the even word of a double word in the IB. The State machine also raises Ca' at this time.
500 nS	Increment fPCd
550 nS	Assuming Respv' is raised within 150 nS of Respu', the buffered version of Respv' will be seen now. This stores the data into the odd word of the double IB word.
600 nS	Set the Full bit corresponding to the IB double word just filled.
650 nS	Return to the idle state (S2). Release EmuClk if it was being held waiting for the IBReference.

John Dillon's preliminary estimate of the Ca' to Resp' time is 195 nS (8/15/83).

His preliminary estimate of the Respu' HI to Respv' HI time is 140 nS (8/15/83).

Note that read data is actually on the AP Bus between 50 and 100 nS before the synchronized version of Respu' and Respv' reach the Bus State Machine. The data is latched in the Simulator's Connector Card. The exact duration of read data on the AP Bus is not known.