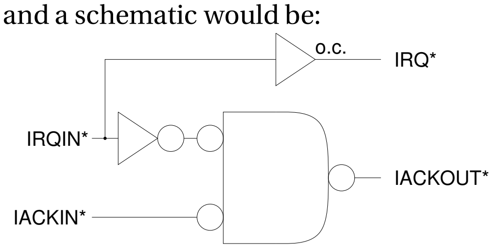
Assignment 7 - Solutions

Question 1

The daisy-chain interrupt acknowledge arbitration circuit asserts IACK-OUT* only if IACKIN* is asserted and the device is not requesting an interrupt itself. The device asserts IRQ* whenever it is requesting an interrupt. If we call the interrupt request input IRQIN*, the truth table (showing truth values) for the arbitration circuit is as follows:

IRQIN*	IACKIN*	IRQ*	IACKOUT*
0	0	0	0
0	1	0	1
1	0	1	0
1	1	1	0

IRQ* thus has the same value as IRQIN* (although it must have an open-collector output), and IACKOUT* can be generated as (in CUPL format) !IRQIN & IACKIN and a schematic would be:



You could, of course, draw the schematic using different components.

Question 2

The period of the 40 MHz clock is 25 ns. If we assume a CPU read requires 5 ns setup and 0 ns hold (reasonable assumptions for this CPU) we can allow one clock cycle (25 ns) for accesses from cache RAM (20 ns access time) and three clock cycles (75 ns) for accessses from the main memory (70 ns access time).

The controller will have three states: an initial state during which all memories are accessed simultaneously. At the end of the first state MISS* will be valid and we decide whether to return to the initial state for another read cycle or continue with two additional wait states to allow sufficient time for access from the main memory.

We enable the cache memory outputs (which could be the chip output enables or the enables of separate tri-state buffers) during the first clock cycle in case there is a cache hit. In case of a miss we disable the cache and enable the main memory in the third clock cycle. Note that access times are measured from address valid to data out valid *not* from output enable to data out valid.

The output enable to data out valid time is simply the time required to turn on the chip's buffers and is typically much shorter than the access time (25 ns would certainly be sufficient).

We assert WAIT* at (or more likely, before) the end of the first clock cycle in case of a cache miss.

We must assert WAIT* during the second and (possibly) third clock cycles. The actual behaviour of the CPU with respect to WAIT* was not well specified in the question so a number of answers are possible. If you assumed that the CPU samples WAIT* at the start of each clock cycle then it is not necessary to hold WAIT* during the third clock cycle. If you assumed it was necessary to hold WAIT* until the end of the read cycle then WAIT* must be asserted during the third clock cycle.

(a)

(b) The truth table for the state machine could be as follows. This state machine assumes the CPU latches WAIT* internally at the start of each clock cycle so that WAIT* need not be enabled during the third clock cycle of a main memory read.

A	B	RD*	MISS*	CACHE*	MAIN*	WAIT*	A'	B'
0	0	0	X	0	0	0	0	0
0	0	1	0	1	0	0	0	0
0	0	1	1	1	0	1	0	1
0	1	X	X	0	0	1	1	0
1	0	X	X	0	1	0	0	0

(c) See attached listings.