Data Sheet For 80186XL Core

DOCUMENT REVISION HISTORY

Revision	Date	Change Description	Author
0.0	22nd March'08	Initial Draft Version	MJ
0.1	18th March'09	Updated for 80186XL	Arun
REL 1.0	29th Nov'11	Updated as per the latest formatting	Arun
REL 1.1	16th Aug'12	Removed implementation results	VC

PROPRIETARY NOTICE: This document contains proprietary material for the sole use of the intended recipient(s). Do not read this document further if you are not the intended recipient. Any review, use, distribution or disclosure by others is strictly prohibited. If you are not the intended recipient (or authorized to receive for the recipient), you are hereby notified that any disclosure, copy or distribution or use of any of the information contained within this document is STRICTLY PROHIBITED. Thank you. "*i*Wave Systems Tech. Pvt. Ltd."

Table of Contents

List Of Figures

List Of Tables

1 Introduction

1.1 Purpose

The iW-80186XL is a powerful 16-bit microprocessor core, executes instruction list compatible with 80186XL microprocessor. The design along with multiple peripherals can be fit into single FPGA.

1.2 Features

The following are the main features of the 80186XL Core:

- iW-80186XL CPU Core
  - Multiplexed 20-bit address and 16-bit data bus
  - 1M-byte memory space divided into 4 segments
  - 64K-byte IO space
  - Non Maskable Interrupt support
  - Arithmetic-Logic Unit
    - 8,16,32-bit arithmetic operations
    - 8,16-bit logical operations
    - Boolean manipulations
    - 16 x 16 bit multiplication (signed or unsigned)
    - 32/16-bit division (signed or unsigned)
- CPU On-Chip Peripherals
  - Programmable Timer / Counter Unit
    - Three programmable independent 16-bit timers
    - TOUT0 to TOUT1 pin outputs
    - TIN0 & TIN1 used either as clock or control signals
    - Timer-2 can be used to clock other two timers
    - Internal / external input clock selectable
  - Direct Memory Access Unit
    - Two independent high-speed DMA channels.
    - Data can be transferred between any combination of memory & IO space.
    - DMA transfer can be initiated by external, internal request or by direct programming.
    - 20-bit length address register.

- 16-bit length transfer count register.
- Transfer address can be incrementing, decrementing or remained constant.
- Two kinds of channel priority order
  - ✓ Fixed priority
  - ✓ Rotating priority
- DMAU can be programmed to produce interrupt request when its transfer count reaches zero.
- Both byte & word transfer is possible in case of 80186XL;word transfer is illegal in case of 80186XL processor.

o Interrupt Control Unit

- Four external interrupt request inputs (INT0 to INT3).
- Timer0, Timer1, Timer2 and DMA0, DMA1 Interrupts (Internal Interrupts)
- Edge or level triggered interrupt request inputs.
- Individually Mask-able interrupts request
- Programmable interrupt request priority orders.
- Polling operation capability.
- Cascade with external 8259A interrupts (only on INT0 and INT1) operates in either Master mode or Slave mode.
- Special fully nested mode support

o Chip Select Unit

- Thirteen programmable chip-select outputs
- Six of the chip-selects map only into memory address space, while the remaining seven can map into either memory or I/O address space
- Programmable block size and start / end address
- Memory or I/O bus cycle decoder
- Programmable wait-state generator
- Provision to disable a chip-select
- Provision to override bus ready

o Clock Generator

1.3 Features Not Supported

o Oscillators internal to the FPGA, so crystals cannot be directly connected to the FPGA

o ONCE mode

- Power down modes
- Refresh Control Unit
- WAIT instruction is not supported

1.4 Acronyms and Abbreviations

Table 1: Acronyms & Abbreviations

Term	Meaning
FPGA	Field Programmable Gate Array
CPU	Central Processing Unit
I/O	Input/ Output

2 80186XL Core

2.1 Block Diagram

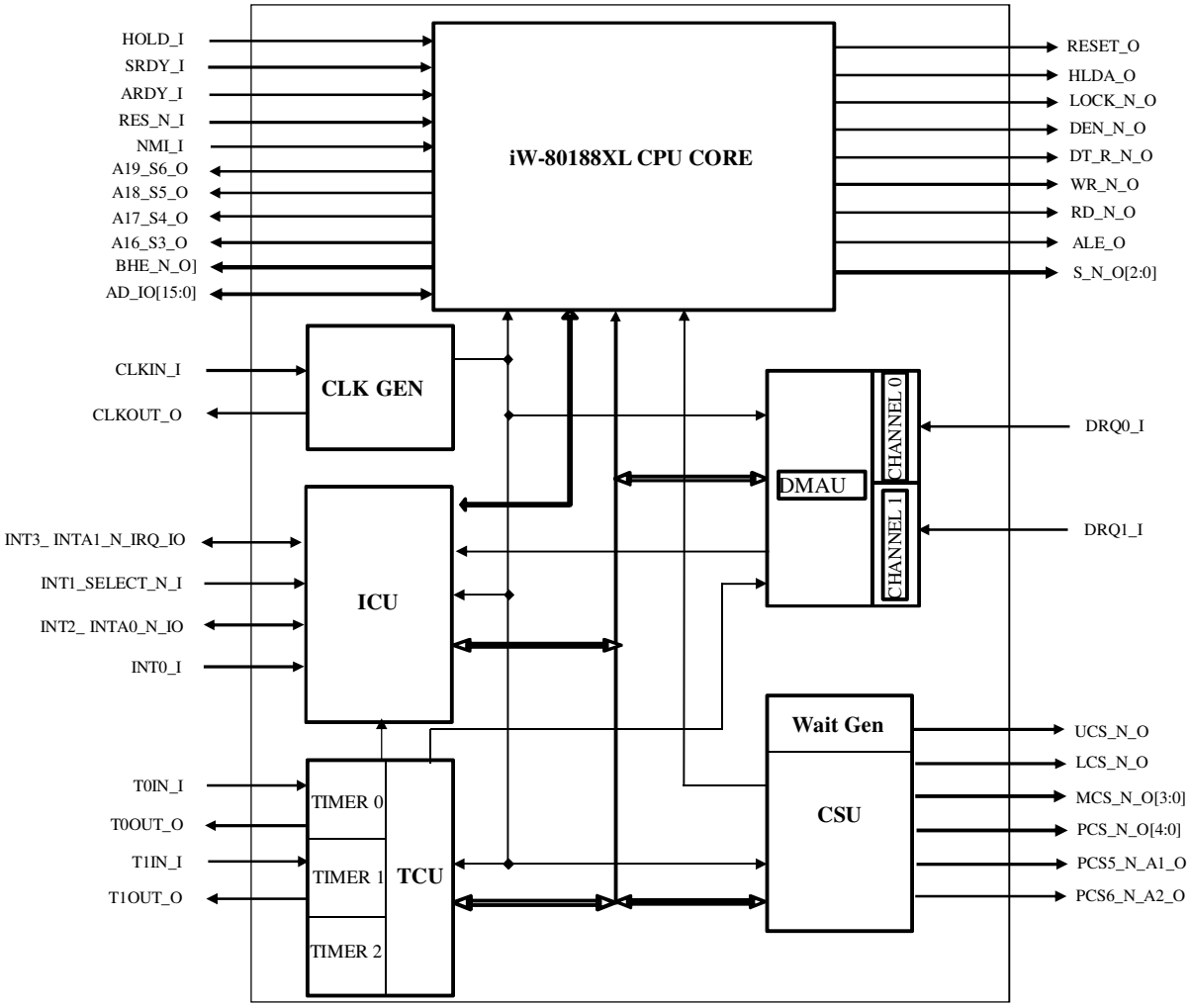

Figure 1: 80186XL Core Block Diagram

2.2 Description

The main blocks in 80186XL Core:

- **Central Processing Unit (CPU)**: This module executes instructions, which include fetching, decoding instructions and generating appropriate requests to the Bus Interface Unit.
- **Timer / Counter Unit (TCU):** This module provides three programmable 16-bit Timers. Two of these are highly flexible and are connected to external pins for control or clocking. A third timer is not connected to any external pins and can only be clocked internally.
- **Direct Memory Access Unit (DMAU):** This module allows data to be transferred between memory and I/O devices without requiring the CPU's intervention. The DMA Unit has two channels. Each channel can accept DMA requests from one of three sources: an external request pin, the Timer/Counter Unit or direct programming. Data can be transferred between any combination of memory and I/O space.
- **Interrupt Control Unit (ICU):** This module processes three internal and four external interrupt requests by allocating a priority level to the each request. The Interrupt Control Unit can independently mask each interrupt source or the CPU can globally mask all interrupts. The Interrupt Control Unit operates in either of two modes: Master or Slave.
    - **Master mode**: The ICU controls the mask-able interrupt input to the CPU. Interrupts can originate from the on-chip peripherals and from four external interrupt pins.
    - **Slave mode**: An external 8259A module controls the mask-able interrupt input to the CPU and acts as the master interrupt controller. The ICU processes only those interrupts from the on-chip peripherals and acts as an interrupt input to the 8259A
- **Chip-Select Unit (CSU):** This module integrates logic, which provides up to thirteen programmable chip selects to access memories and peripherals. Six of the chip-selects map only into memory address space, while the remaining seven can map into either memory or I/O address space. Besides selecting a specific device, each chip-select can be used to control the number of wait states inserted into the bus cycle
- **Clock Generator:** This module generates both internal and external clock

2.3 I/O Signal Description

Table 2: 80186XL Core IO Signals

<table>
<tr><th>Signal</th><th>I/O</th><th>Width</th><th>Description</th></tr>
<tr><td>RES_N_I</td><td>I</td><td>1</td><td>An active low signal causes the processor to immediately terminate any bus cycle in progress and assume an initialized state. All pins will be driven to a known state, and RESET will also be driven active.</td></tr>
<tr><td>RESET_O</td><td>O</td><td>1</td><td>Indicates the processor is currently in the reset state. RESET will remain active as long as RES_N remains active.</td></tr>
<tr><td>CLKIN_I</td><td>I</td><td>1</td><td>External clock input operating at two times the processor operating frequency.</td></tr>
<tr><td>CLKOUT_O</td><td>O</td><td>1</td><td>Processor Clock output. It is half of Clock input (CLKIN_I).</td></tr>
<tr><td>AD_IO[15:0]</td><td>IO</td><td>16</td><td>Provide a multiplexed address and Data bus. During the address phase of the bus cycle, address bits AD [15:0] are presented on the bus and can be latched using ALE. 16-bit data information is transferred during the data phase of the bus cycle</td></tr>
<tr><td>BHE_N_O</td><td>O</td><td>1</td><td>Byte High Enable output indicates that the bus cycle in progress is transferring data over the upper half of the data bus. BHE and A0 have the following logical encoding scheme
<table>
<tr><th>A0</th><th>BHE_N_O</th><th>Encoding</th></tr>
<tr><td>0</td><td>0</td><td>Word Transfer</td></tr>
<tr><td>0</td><td>1</td><td>Even byte transfer</td></tr>
<tr><td>1</td><td>0</td><td>Odd byte transfer</td></tr>
<tr><td>1</td><td>1</td><td>NA</td></tr>
</table></td></tr>
<tr><td>A19_S6_O,
A18_S5_O,
A17_S4_O,
A16_S3_O,</td><td>O</td><td>1</td><td>Address Bus Outputs and Bus Cycle Status (3-6) indicate the four most significant address bits during T1. These signals are active HIGH.
During T2, T3, TW and T4, the S6 pin is LOW to indicate a CPU-initiated bus cycle or HIGH to indicate a DMA initiated. During the same T-states, S3, S4 and S5 are always LOW.</td></tr>
<tr><td>ALE_O</td><td>O</td><td>1</td><td>Address Latch Enable, an active high signal used to strobe address information into a transparent type latch during the address phase of the bus cycle.</td></tr>
</table>

<table>
<tr><th>Signal</th><th>I/O</th><th>Width</th><th>Description</th></tr>
<tr><td>S_N_O[2:0]</td><td>O</td><td>3</td><td>Bus cycle Status are encoded on these pins to provide bus transaction information. S[2:0] are
<table>
<tr><th>S2</th><th>S1</th><th>S0</th><th>BUS CYCLE</th></tr>
<tr><td>0</td><td>0</td><td>0</td><td>Interrupt Acknowledge</td></tr>
<tr><td>0</td><td>0</td><td>1</td><td>Read I/O</td></tr>
<tr><td>0</td><td>1</td><td>0</td><td>Write I/O</td></tr>
<tr><td>0</td><td>1</td><td>1</td><td>Processor HALT</td></tr>
<tr><td>1</td><td>0</td><td>0</td><td>Queue Instruction Fetch</td></tr>
<tr><td>1</td><td>0</td><td>1</td><td>Read Memory</td></tr>
<tr><td>1</td><td>1</td><td>0</td><td>Write Memory</td></tr>
<tr><td>1</td><td>1</td><td>1</td><td>Passive(no bus activity)</td></tr>
</table></td></tr>
<tr><td>ARDY_I</td><td>I</td><td>1</td><td>Asynchronous Ready informs the processor that the addressed memory space or I/O device will complete a data transfer. The ARDY pin accepts a rising edge that is asynchronous to CLKOUT and is active HIGH. The falling edge of ARDY must be synchronized to the processor clock. Connecting ARDY HIGH will always assert the ready condition to the CPU. If this line is unused, it should be tied LOW to yield control to the SRDY pin.</td></tr>
<tr><td>SRDY_I</td><td>I</td><td>1</td><td>Synchronous Ready informs the processor that the addressed memory space or I/O device will complete a data transfer. The SRDY pin accepts an active-HIGH input synchronized to CLKOUT. The use of SRDY allows relaxed system timing over ARDY. This is accomplished by elimination of the one-half clock cycle required to internally synchronize the ARDY input signal.</td></tr>
<tr><td>RD_N_O</td><td>O</td><td>1</td><td>Active low Read output signals that the accessed memory or I/O device must drive data information onto the data bus.</td></tr>
</table>

Signal	I/O	Width	Description
WR_N_O	O	1	Active low Write output signals that data available on the data bus are to be written into the accessed memory or I/O device.
LOCK_N_O	O	1	The processor will not service other bus requests (such as HOLD) while LOCK is active.
DT_R_N_O	O	1	Data Transmit/Receive controls the direction of data flow through an external data bus transceiver. When LOW, data is transferred to the processor. When HIGH the processor places write data on the data bus.
DEN_N_O	O	1	Data Enable is provided as a data bus transceiver output enable. DEN_N is active LOW during each memory and I/O access. DEN_N is HIGH whenever DT/R_N changes state. During RESET, DEN_N is driven HIGH for one clock, then floated.
HOLD_I	I	1	HOLD request input to signal that an external bus master wishes to gain control of the local bus.
HLDA_O	O	1	The processor generates HLDA in response to a HOLD indicating that bus is granted. It indicates that the processor has relinquished control of the local bus.
NMI_I	I	1	Non-Maskable Interrupt input causes a TYPE-2 interrupt to be serviced by the CPU. NMI is latched internally.
DRQ0_I, DRQ1_I	I	1	DMA Request is asserted HIGH by an external device when it is ready for DMA Channel 0 or Channel 1 to perform a transfer. These signals are level-triggered and internally synchronized.
INT0_I, INT1_SELECT_N_I,	I	1	Maskable Interrupt Requests can be requested by activating one of these pins. Interrupt Requests are synchronized internally. INT1 is multiplexed with Slave SELECT input to the ICU (Slave mode).

Signal	I/O	Width	Description
INT2_INTA0_N_IO, INT3_INTA1_N_IRQ_IO	IO	1	Maskable Interrupt Requests can be requested by activating one of these pins. When configured as inputs, these pins are active HIGH. Interrupt Requests are synchronized internally. INT2 and INT3 may be configured to provide active-LOW interrupt acknowledge output signals. When Slave Mode is selected, Output is Interrupt Request (IRQ) from the Controller to the external 8259A.
T0OUT0_O, T1OUT1_O	O	1	Timer output pins can be programmed to provide a single clock or continuous waveform generation, depending on the timer mode selected.
T0IN0_I, T1IN1_I	I	1	Timer input is used either as clock or control signals, depending on the timer mode selected.
UCS_N_O	O	1	Upper Memory Chip Select is an active LOW output whenever a memory reference is made to the defined upper portion (1K-256K block) of memory. The address range activating UCS_N is software programmable.
LCS_N_O	O	1	Lower Memory Chip Select is active LOW whenever a memory reference is made to the defined lower portion (1K-256K) of memory. The address range activating LCS_N is software programmable.
MCS_N_O [3:0]	O	4	Mid-Range Memory Chip Select signals are active LOW when a memory reference is made to the defined midrange portion of memory (8K-512K). The address ranges activating MCS_N [0-3] are software programmable.
PCS_N_O [4:0]	O	5	Peripheral Chip Select signals are active LOW when a reference is made to the defined peripheral area (64 Kbyte I/O or 1 MByte memory space). The address ranges activating PCS_N [0-4] are software programmable.
PCS5_N_A1_O	O	1	Peripheral Chip Select 5 or Latched A1 may be programmed to provide a sixth peripheral chip select, or to provide an internally latched A1 signal. The address range activating PCS5_N is software-programmable.

Signal	I/O	Width	Description
PCS56_N_A2_O	O	1	Peripheral Chip Select 6 or Latched A2 may be programmed to provide a seventh peripheral chip select, or to provide an internally latched A2 signal. The address range activating PCS6_N is software-programmable.

3 Timing Waveforms

3.1 CPU Write Cycle

3.1.1 CPU Write Cycle (BUS_RDY_DISABLE = 1 and Without wait state)

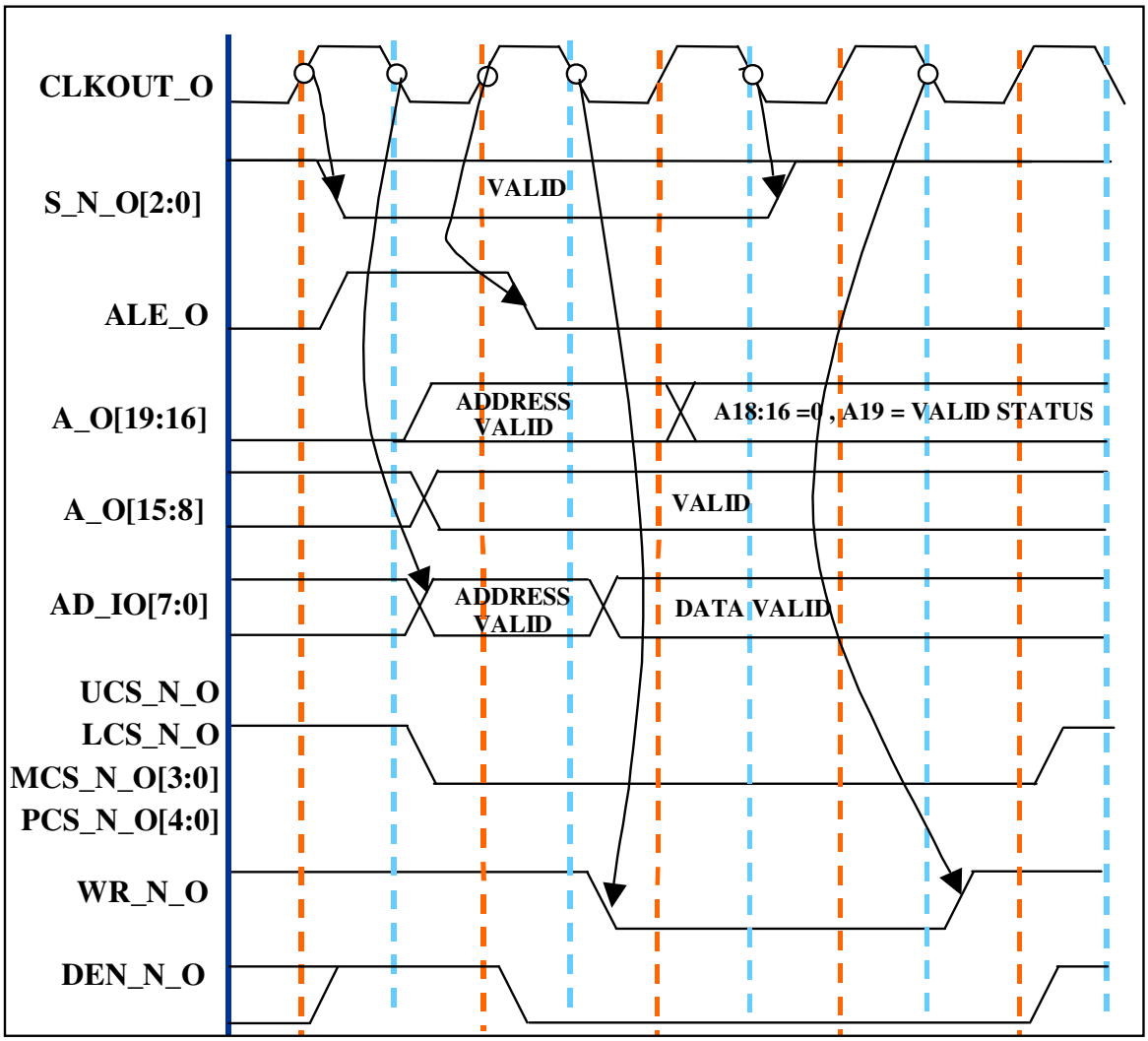

Figure 2: CPU Write Cycle (BUS_RDY_DISABLE = 1 and Without wait state)

3.1.2 CPU Write Cycle (BUS_RDY_DISABLE = 1 and With wait state)

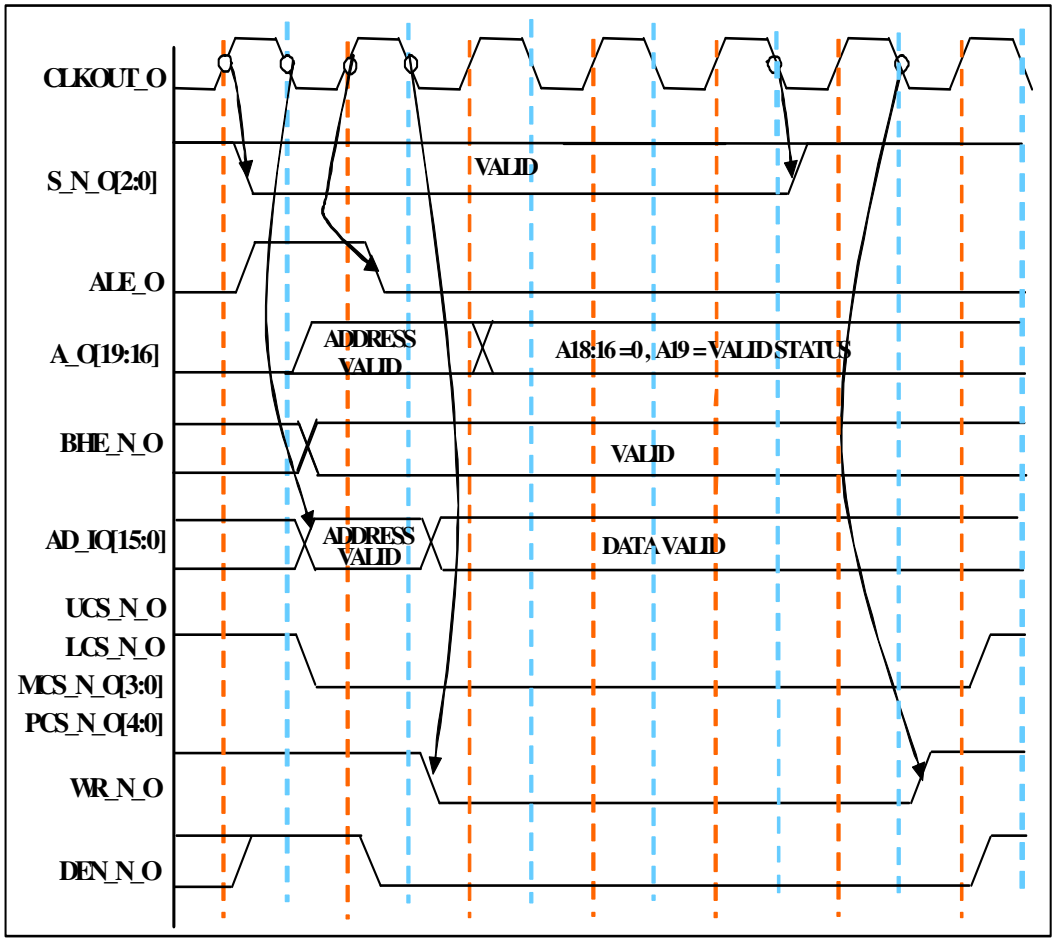

Figure 3: CPU Write Cycle (BUS_RDY_DISABLE = 1 and With wait state)

3.1.3 CPU Write Cycle (BUS_RDY_DISABLE = 0)

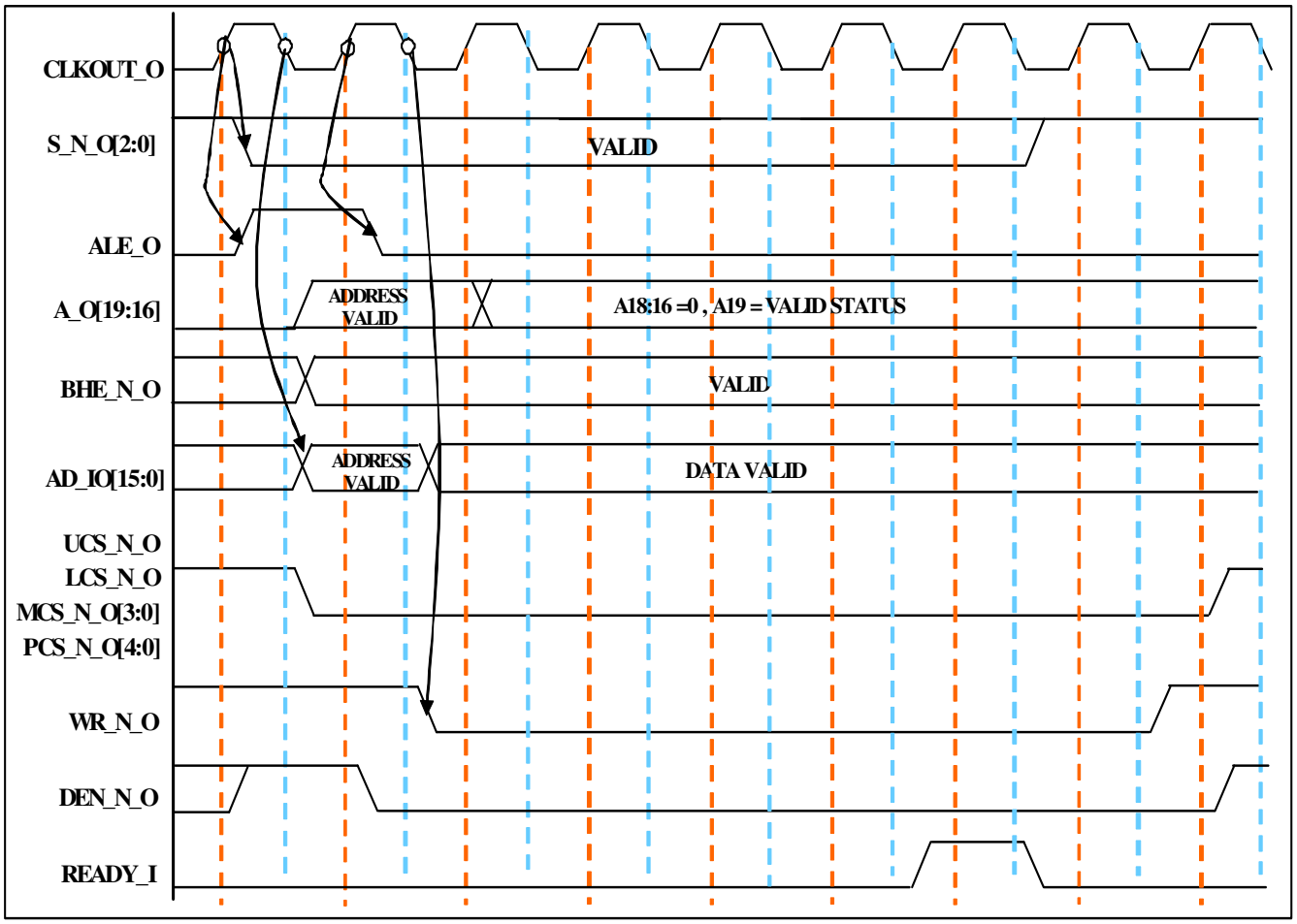

Figure 4: CPU Write Cycle (BUS_RDY_DISABLE = 0)

3.2 CPU Read Cycle

3.2.1 CPU Read Cycle

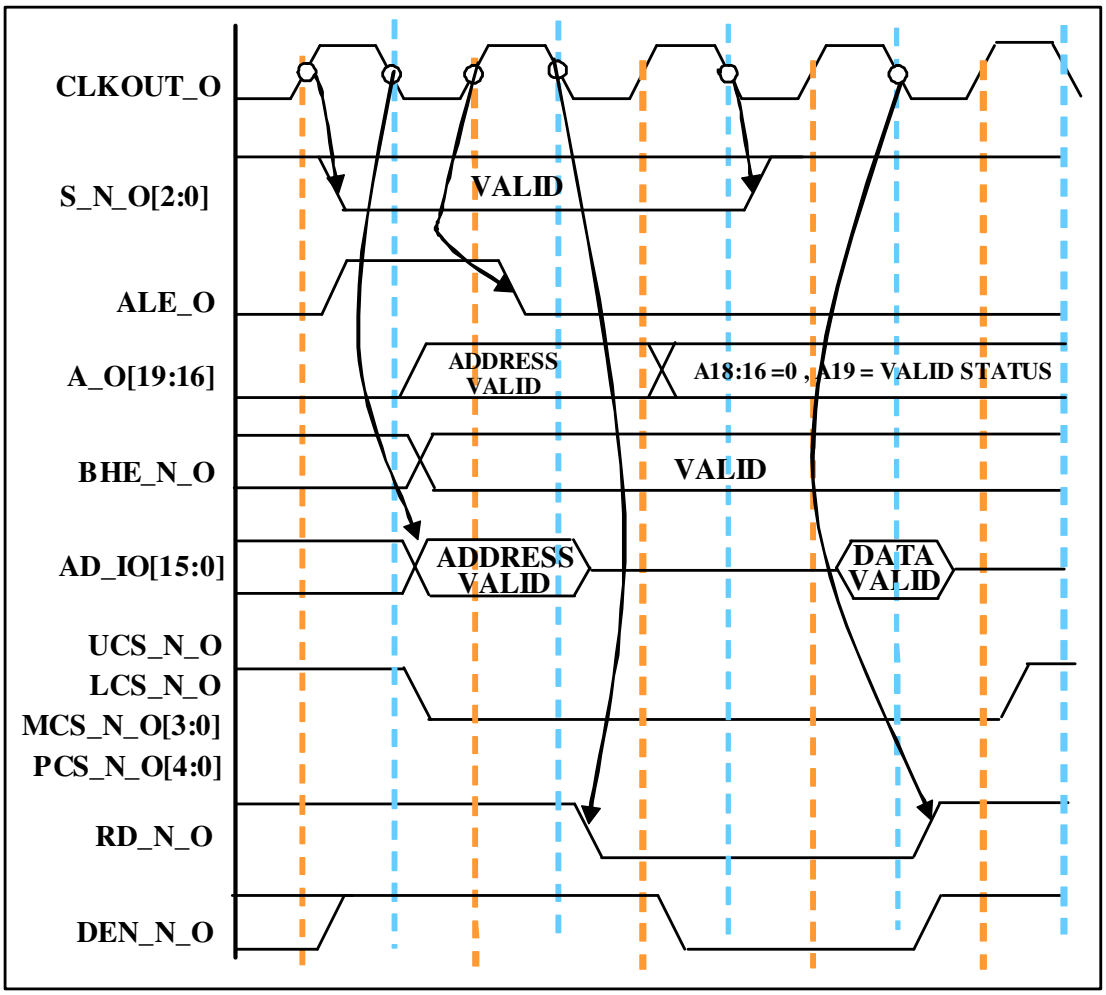

Figure 5: CPU Read Cycle

3.3 Interrupt Acknowledge Cycle

3.3.1 Interrupt Acknowledge Cycle

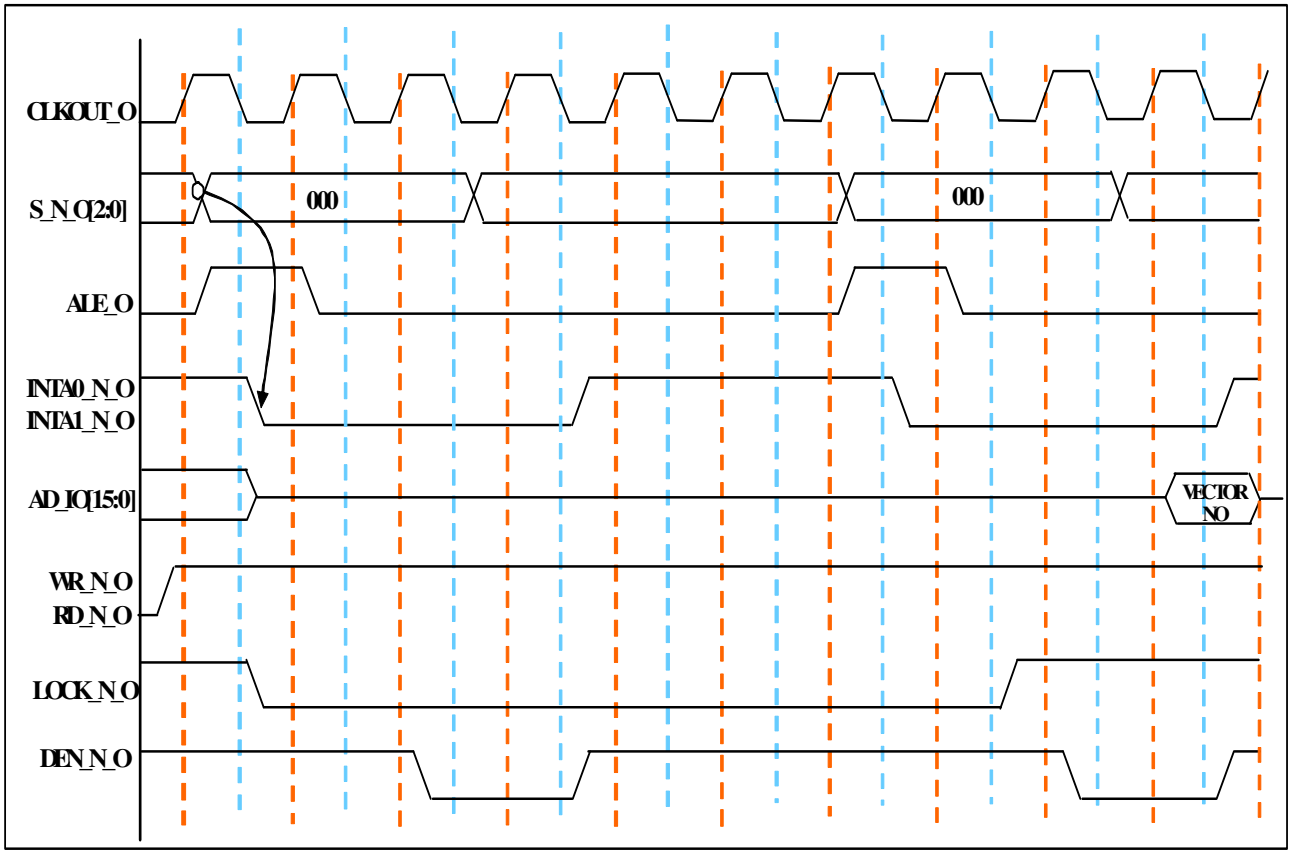

Figure 6: Interrupt Acknowledge Cycle

3.4 HOLD/HACK Cycle

3.4.1 HOLD/HACK Cycle

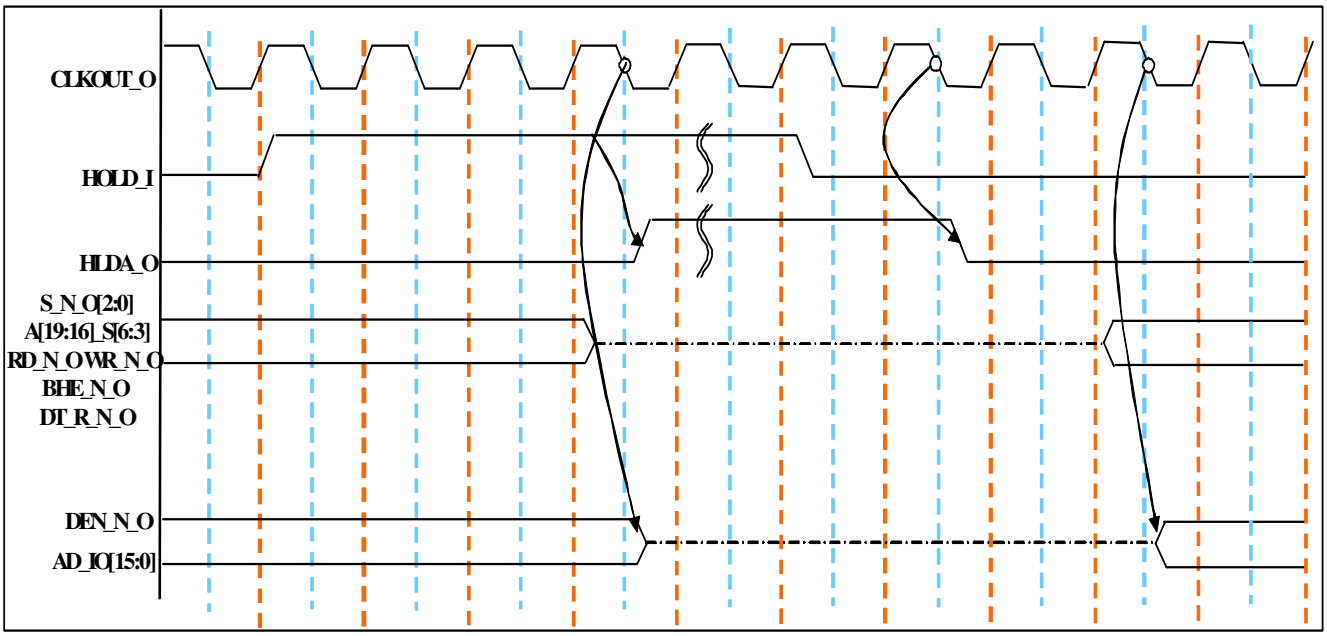

Figure 7: HOLD/HACK Cycle

3.5 HALT Cycle

3.5.1 HALT Cycle

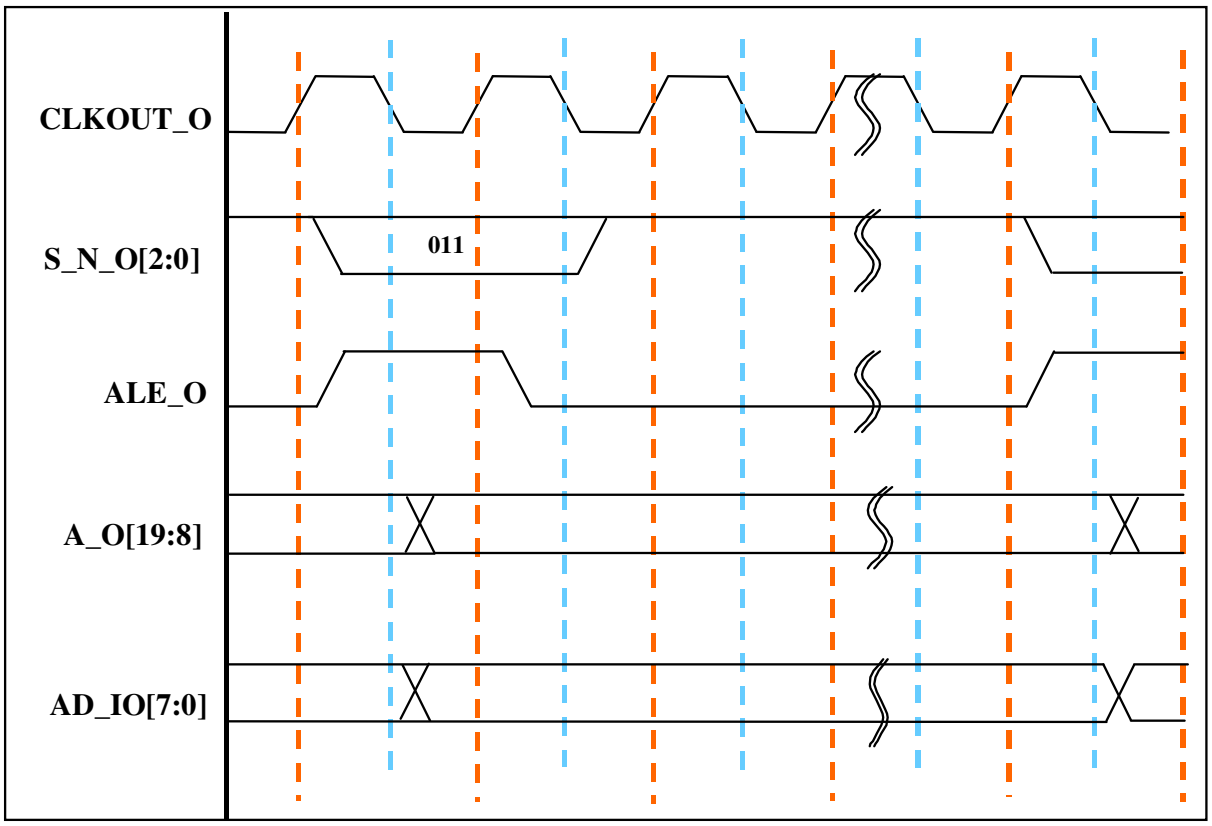

Figure 8: HALT Cycle