



TLA700 Series Target Signal Requirements For ARM Microprocessor Support Packages

Dragonfly Software Development LLC
8196 SW Hall Blvd, Suite 104
Beaverton, OR 97008

www.dfsw.com
support@dfs.com

Document Version 1.02

Table of Contents

1. INTRODUCTION.....	1
2. ACQUIRING SAMPLES FROM THE BUS.....	1
2.1. CLOCKING MECHANISMS.....	1
2.1.1. <i>Internal Clocking</i>	1
2.1.2. <i>External Clocking</i>	1
2.1.3. <i>Custom Clocking</i>	2
2.2. CHOOSING A CLOCKING METHOD.....	2
3. AMBA-ASB MICROPROCESSOR SUPPORT PACKAGE	3
3.1. OVERVIEW.....	3
3.2. SUMMARY OF SIGNAL REQUIREMENTS.....	3
3.3. REQUIRED SIGNALS FOR ACQUISITION.....	4
3.4. REQUIRED SIGNALS FOR DISASSEMBLY.....	4
3.4.1. <i>Address Bus (BA[31:0])</i>	4
3.4.2. <i>Data Bus (BD[31:0])</i>	4
3.4.3. <i>Bus Control Signals</i>	4
4. ARM7T MICROPROCESSOR SUPPORT PACKAGE	6
4.1. OVERVIEW.....	6
4.2. SUMMARY OF SIGNAL REQUIREMENTS.....	6
4.3. REQUIRED SIGNALS FOR ACQUISITION.....	7
4.4. REQUIRED SIGNALS FOR DISASSEMBLY.....	7
4.4.1. <i>Address Bus (A[31:0])</i>	7
4.4.2. <i>Data Bus (D[31:0])</i>	7
4.4.3. <i>Bus Control Signals</i>	7

1. Introduction

The TLA700 ARM Microprocessor Support Packages offer timing analysis, state analysis and both ARM and Thumb disassembly support for ARM microprocessors utilizing either the ARM7T or AMBA-ASB bus. To effectively utilize the TLA700 ARM Microprocessor Support Packages, the TLA700 must be able to monitor both the address and data bus of the microprocessor, as well as several of the bus control signals, in order to properly decode the activity of the bus.

However, since the ARM microprocessor core is typically embedded within an application-specific ASIC, not all of the signals from the core will be available for monitoring, due to pinout limitations on the ASIC, or other application-dependent issues.

Both the ARM7T and AMBA-ASB microprocessor support packages can be utilized in the absence of the full set of the microprocessor bus control signals, though in some cases the utility of the package will be diminished. This document describes the bus signal requirements of both the ARM7T and AMBA-ASB Microprocessor Support Packages, and describes the limitations which are imposed on the package when not all bus control signals are available.

2. Acquiring Samples from the Bus

Each TLA700 Microprocessor Support Package can be logically divided into two components. The first component, the "clocking state machine", is used to recognize notable events that occur on the ARM bus, so that these events can be stored into the acquisition memory of the logic analyzer. The second component, the disassembler, is used to post-process the contents of the logic analyzer's acquisition memory, in order to provide a listing of the events that have been captured. Several clocking mechanisms are available, and can be utilized with either ARM support package.

2.1. Clocking Mechanisms

In addition to the specialized clocking state machines which are developed specifically for each Microprocessor Support Package, the TLA700 provides built-in clocking support for both "internal" and "external" clocks. Each of these clocking mechanisms can be utilized to acquire activity from the ARM bus.

2.1.1. Internal Clocking

With Internal Clocking, the state of each monitored signal is sampled repeatedly, at a user-selectable time interval. This clocking mechanism allows the activity of the monitored signals to be recorded, even in the absence of a clock signal from the bus. However, since the clocking of the bus into the logic analyzer's acquisition memory is not synchronized with the activity of the bus, this type of acquisition is inherently "noisy". In addition, it is possible to miss relevant bus activity if the time interval between acquisitions is too large.

2.1.2. External Clocking

With External Clocking, it is possible to logically combine several user-specified clock and qualifier signals from the system under test to control when samples are transferred to acquisition memory. In most cases, the signals which would be used to control external clocking can be used instead by the custom clocking state machines provided by the support packages. As a result, External Clocking is seldom used with the support packages.

2.1.3. Custom Clocking

The ARM microprocessor support packages each provide customized clocking mechanisms which are tailored to the activity of the ARM bus. By tailoring the clocking state machine to the specific control signals of the bus, the amount of wasted acquisition memory can be kept to a minimum, since extra bus cycles for wait states, etc., can be eliminated from the acquisition, thus increasing the amount of relevant bus activity which can be captured.

2.2. *Choosing a Clocking Method*

The Internal, External and Custom clocking mechanisms can each be used to control when the TLA700 acquires samples from the microprocessor bus. Ideally, one of the Custom clocking mechanisms offered by the Microprocessor Support package should be utilized, both to minimize the amount of acquisition memory which is utilized, and to reduce the amount of superfluous information which is presented by the disassembler listing. To utilize the Custom clocking features of the package, the signals required for the desired custom clocking method must be available to the TLA700.

3. AMBA-ASB Microprocessor Support Package

3.1. Overview

The AMBA-ASB Microprocessor Support Package provides timing analysis, state analysis and both ARM and Thumb disassembly support for bus cycles which are acquired from the AMBA-ASB bus. The package provides several Custom clocking options which can be utilized to tailor the acquisition of signals to the specific requirements of the system under test.

3.2. Summary of Signal Requirements

The AMBA-ASB Microprocessor Support Package can make use of the following bus signals if they are available to the logic analyzer:

SIGNAL	REQUIRED?	EFFECT IF SIGNAL(S) NOT AVAILABLE
BCLK	Strongly Recommended	Internal or External clocking must be used to acquire bus cycles.
BWAIT	Strongly Recommended	Bus wait state cycles will be acquired and disassembled, even though the bus cycle is not appropriate for disassembly.
BTRAN[1]	Recommended	Address-only bus cycles will be acquired and disassembled, even though the bus cycle is not appropriate for disassembly.
BD[31-0]	Yes	No disassembly.
BA[31-2]	Optional	Disassembled address values will be incorrect.
BA[1]	Strongly Recommended	Disassembly of Thumb instructions and display of 16-bit data transfers will be inaccurate.
BA[0]	Recommended	Display of 8-bit data transfers will be inaccurate.
Bsize[1:0]	Strongly Recommended	ARM/Thumb disassembly support is degraded, as is identification of data transfer sizes.
BWRITE	Recommended	Data transfer direction will be misidentified.
BPROT[1]	Optional	No impact on disassembly.
BPROT[0]	Strongly Recommended	No automatic detection of instruction fetch bus cycles.

3.3. Required Signals for Acquisition

The AMBA-ASB bus signals can be acquired using Internal, External and Custom Clocking. Three Custom Clocking options are provided with the support package:

- Both Edges of BCLK
- Simplified ASB Cycles
- Valid ASB Cycles

The following signals are required for each clocking method:

Custom Clocking Option	BCLK	BWAIT	BTRAN[1]
Both Edges of BCLK	X		
Simplified ASB Cycles	X	X	
Valid ASB Cycles	X	X	X

By utilizing the BWAIT signal, both the "Simplified ASB" and "Valid ASB Cycles" clocking methods suppress the additional wait states which will be present during access to slow bus devices. By utilizing the BTRAN[1] signal, the "Valid ASB Cycles" clocking method suppresses address-only bus cycles which are sometimes generated by the microprocessor.

3.4. Required Signals for Disassembly

Full disassembly support for AMBA-ASB requires that the address and data bus signals be available to the TLA700, as well as a subset of the AMBA-ASB bus control signals. The support package can still function, though with reduced functionality, if some of these signals are not available.

3.4.1. Address Bus (BA[31:0])

The address bus signals are used to determine which data bus byte lanes are used for 16-bit and 8-bit data transfers. If BA[1] is not available, the disassembler will be unable to determine which portion of the data bus to use to disassemble Thumb instructions. If Thumb disassembly is required, BA[1] should be available to the analyzer. If BA[0] is not available, the disassembler will be unable to determine which portion of the data bus to use to display 8-bit data transfers.

Unused address bus signals need not be connected to the analyzer. They may be tied to ground, or left unconnected, in which case they will be pulled high by the logic analyzer.

3.4.2. Data Bus (BD[31:0])

All 32 bits of the data bus are used by the disassembler for ARM instruction decoding, and must be available for disassembly.

3.4.3. Bus Control Signals

The following AMBA-ASB bus signals are collectively referred to as the Control group by the AMBA-ASB support package:

- BWRITE
- BSIZE[1:0]
- BPROT[1:0]

Along with the address and data bus signals, these control signals are used by the support package during disassembly.

BWRITE

The BWRITE signal is used by the support package to differentiate between data read and data write operations. If this signal is absent, the support package will be unable to differentiate between data read and data write operations. Disassembly of ARM and Thumb instructions is unaffected.

BSIZE[1:0]

The BSIZE[1:0] signals are used by the support package to determine the size of the data transfer for the acquired bus cycle. For data read and write operations, these signals are used to identify the size of the data being read or written. For instruction fetch operations, BSIZE[1:0] are used to determine whether or not the instruction fetch refers to an ARM or Thumb instruction. In the absence of either BSIZE[1] or BSIZE[0], the missing signal(s) can be forced to either a high or low state, allowing the disassembler to decode the cycle using the following assumptions:

BSIZE State	Instruction Fetch	Data Transfer
BSIZE[1:0] = 0 0	Cycle type marked as <i>unknown</i>	Assume 8-bit transfer
BSIZE[1:0] = 0 1	Assume Thumb Instruction	Assume 16-bit transfer
BSIZE[1:0] = 1 0	Assume ARM Instruction	Assume 32-bit transfer
BSIZE[1:0] = 1 1	Assume ARM Instruction	Cycle type marked as <i>unknown</i>

BPROT[1:0]

The BPROT[1:0] signals are used to identify the type of cycle associated with the acquired sample. If present, BPROT[1] is used to distinguish between user and supervisor bus cycles, but disassembly performance is unaffected by its absence. BPROT[0] is used to distinguish between instruction and data fetches. In the absence of BPROT[0], this signal can be forced to either a high or low state, allowing the disassembler to decode the cycle using the following assumptions:

BPROT[0] = 0	Cycle assumed to be an instruction fetch
BPROT[0] = 1	Cycle assumed to be a data fetch

Cycles which are incorrectly labeled as data fetches by the disassembler can be manually marked by the user as instruction fetches, allowing the disassembler to provide instruction decoding for the cycle.

4. ARM7T Microprocessor Support Package

4.1. Overview

The ARM7T Microprocessor Support Package provides timing analysis, state analysis and both ARM and Thumb disassembly support for bus cycles which are acquired from the ARM7T bus. The package provides several Custom clocking options which can be utilized to tailor the acquisition of signals to the specific requirements of the system under test.

4.2. Summary of Signal Requirements

The ARM7T Microprocessor Support Package can make use of the following bus signals if they are available to the logic analyzer.

SIGNAL	REQUIRED?	EFFECT IF SIGNAL(S) NOT AVAILABLE
MCLK	Strongly Recommended	Internal or External clocking must be used to acquire bus cycles.
nWAIT	Strongly Recommended	Bus wait states will be acquired and disassembled, even though the bus cycle is not appropriate for disassembly.
nMREQ	Recommended	Internal and coprocessor cycles will be acquired and disassembled, even though the bus cycle is not appropriate for disassembly.
D[31-0]	Yes	No disassembly.
A[31-2]	Optional	Disassembled address values will be incorrect.
A[1]	Strongly Recommended	Disassembly of Thumb instructions and display of 16-bit data transfers will be inaccurate.
A[0]	Recommended	Display of 8-bit data transfers will be inaccurate.
MAS[1:0]	Strongly Recommended	ARM/Thumb disassembly support is degraded, as is identification of data transfer sizes.
nRW	Recommended	Data transfer direction misidentified.
nOPC	Strongly Recommended	No automatic detection of instruction fetch bus cycles.

4.3. Required Signals for Acquisition

The ARM7T bus signals can be acquired using Internal, External and Custom Clocking. Three Custom Clocking options are provided with the support package:

- Both Edges of MCLK
- Simplified ARM7T Cycles
- Simplified PID7T Cycles
- Valid ARM7T Cycles
- Valid PID7T Cycles

The following signals are required for each clocking method:

Custom Clocking Option	MCLK	nWAIT	nMREQ
Both Edges of MCLK	X		
Simplified ARM7T Cycles	X	X	
Simplified PID7T Cycles	X	X	
Valid ARM7T Cycles	X	X	X
Valid PID7T Cycles	X	X	X

By utilizing the nWAIT signal, both the "Simplified" and "Valid" clocking methods suppress the additional wait states which are sometimes present during access to slow bus devices. By utilizing the nMREQ signal, the "Valid" clocking methods suppress the address-only and coprocessor bus cycles which are sometimes generated by the microprocessor.

4.4. Required Signals for Disassembly

Full disassembly support for ARM7T requires that the address and data bus signals be available to the TLA700, as well as a subset of the ARM7T bus control signals. The support package can still function, though with reduced functionality, if some of these signals are not available.

4.4.1. Address Bus (A[31:0])

The address bus signals are used to determine which data bus byte lanes are used for 16-bit and 8-bit data transfers. If A[1] is not available, the disassembler will be unable to determine which portion of the data bus to use to disassemble Thumb instructions. If Thumb disassembly is required, A[1] should be available to the analyzer. If A[0] is not available, the disassembler will be unable to determine which portion of the data bus to use to display 8-bit data transfers.

Unused address bus signals need not be connected to the analyzer. They may be tied to ground, or left unconnected, in which case they will be pulled high by the logic analyzer.

4.4.2. Data Bus (D[31:0])

All 32 bits of the data bus are used by the disassembler for ARM instruction decoding, and must be available for disassembly.

4.4.3. Bus Control Signals

The following ARM7T bus signals are collectively referred to as the Control group by the ARM7T support package:

- nRW
- MAS[1:0]
- nOPC

Along with the address and data bus signals, these control signals are used by the support package during disassembly.

nRW

The nRW signal is used by the support package to differentiate between data read and data write operations. If this signal is absent, the support package will be unable to differentiate between data read and data write operations. Disassembly of ARM and Thumb instructions is unaffected.

MAS[1:0]

The MAS[1:0] signals are used by the support package to determine the size of the data transfer for the acquired bus cycle. For data read and write operations, these signals are used to identify the size of the data being read or written. For instruction fetch operations, MAS[1:0] are used to determine whether or not the instruction fetch refers to an ARM or Thumb instruction. In the absence of either MAS[1] or MAS[0], the missing signal(s) can be forced to either a high or low state, allowing the disassembler to decode the cycle using the following assumptions:

MAS State	Instruction Fetch	Data Transfer
MAS[1:0] = 0 0	Cycle type marked as <i>unknown</i>	Assume 8-bit transfer
MAS[1:0] = 0 1	Assume Thumb Instruction	Assume 16-bit transfer
MAS[1:0] = 1 x	Assume ARM Instruction	Assume 32-bit transfer

nOPC

The nOPC signal is used to identify the type of cycle associated with the acquired sample. If present, nOPC is used to distinguish instruction and data fetches. In the absence of nOPC, this signal can be forced to either a high or low state, allowing the disassembler to decode the cycle using the following assumptions:

nOPC = 0	Cycle assumed to be an instruction fetch
nOPC = 1	Cycle assumed to be a data fetch

Cycles which are incorrectly labeled as data fetches by the disassembler can be manually marked by the user as instruction fetches, allowing the disassembler to provide instruction decoding for the cycle.