Application Note, V 1.1, June 2004

AP32065

Memory Access Time in TriCore® 1 TC1M Based Systems

TC1M

Microcontrollers

Never stop thinking

Memory Access Time in TriCore® 1 TC1M Based Systems

Revision History:	**2004-06**	V 1.1
Previous Version:-		
Page	Subjects (major changes since last revision)	
	Minor changes	

We Listen to Your Comments

Any information within this document that you feel is wrong, unclear or missing at all?
Your feedback will help us to continuously improve the quality of this document.
Please send your proposal (including a reference to this document) to:

ipdoc@infineon.com

1 Memory Access Time

1.1 Scope

This application note examines the memory hierarchy in a Tricore®1 TC1M-based system and looks at the access time of each hierarchical level. TC1M is a licensable core, implementing version 1.3 of the TriCore architecture.

Because there are many variables that may influence the access time in a real system, some assumptions have been necessary for this document. These assumptions are listed in each subsection, as appropriate.

1.2 Memory Hierarchy

The memory hierarchy in an embedded system can be considered as a series of levels, with each level producing a different access time.

One way to identify the levels in a specific system is by counting the number of bridges that need to be crossed in order to reach the memory. This approach takes into consideration the fact that in a large number of systems there is also a hierarchy of buses:

- A fast bus close to the processor
- A slower bus used to connect peripherals
- An even slower bus (or buses) to connect to external devices

A possible bus implementation is shown below:

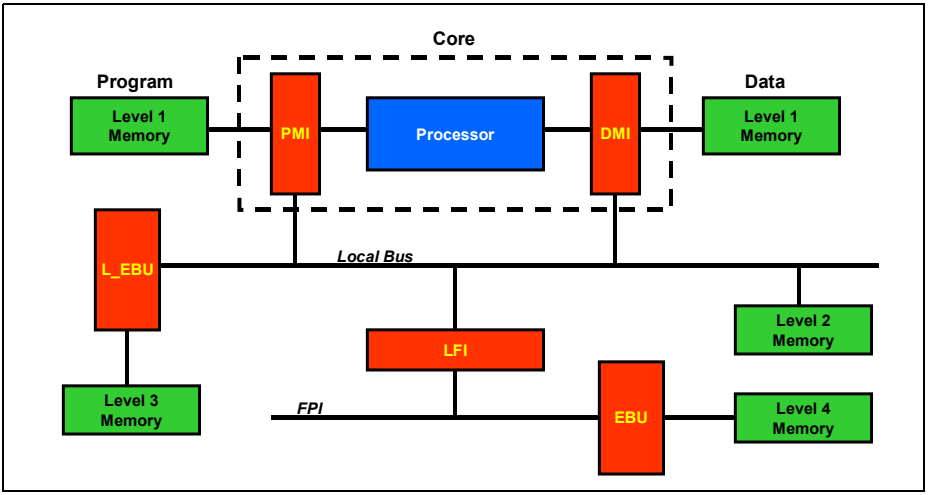

Figure 1 Bus Implementation Example

The following points should be noted about the **Figure 1 Bus Implementation Example** example:

1. The Program Memory Interface (PMI) and Data Memory Interface (DMI) act as the link between the processor and the local bus. They therefore perform a dual function, as both memory interfaces and as bridges to the local bus.

2. As per many existing implementations, the memory closer to the processor (sometimes referred to as internal memory) follows the Harvard architecture, with separate buses for instruction and data.

3. The level of a memory (L) is determined by the simple formula:
   L = *n* +1
   where ***n*** is the number of bridges data has to cross to reach the processor

In **Figure 1 Bus Implementation Example**, data moving in to and out from the memory connected directly to the PMI and DMI does not cross any bridges, and is therefore Level 1 (L1).

Data moving in to and out from the memory connected to the Local Bus needs to cross one bridge (inside the PMI/DMI), and is therefore Level 2 (L2).

Data moving in to and out from the memory connected to the Peripheral Bus needs to cross over two bridges and is therefore Level 3 (L3).

The **Figure 1** block diagram reflects the vast majority of present implementations where the Level 1 memory uses a Harvard structure, while the rest of the memory uses a unified approach. The control of the blocks includes the respective interface circuitry, Program Memory Interface (PMI) and Data Memory Interface (DMI). The core consists of the Processor and the Program Memory Interface (PMI) and the Data Memory Interface (DMI).

The access levels are as follows:

- Level 1 (L1) - memory that can be accessed in one clock cycle (at processor speed)
- Level 2 (L2) - memory that can be accessed in two or more clock cycles
- Level 3/4 (L3/L4) - memory that can be accessed at the slowest speed

Note: The exact number of cycles for level 2, 3 and 4 memories is implementation dependent.

Not all the levels need to be present in a system. L1 and L2 memories are usually internal to the SOC (System-On-a-Chip), while L3 and L4 are external.

1.3 Memory Access Timing

Memory access is given in processor cycles, with the following assumptions:

- LMB and FPI run at the same speed
- The intrinsic access time for memory is one clock cycle

- On each bus, the requestor is already the master, so there is no need for arbitration

1.3.1 L1 Access

Level 1 (L1) memory is accessed in one (1) cycle

1.3.2 L2 Access

Level 2 (L2) memory is assumed to have the same width as the LMB (64b).

1.3.2.1 PMU Access

PMU always fetches a minimum 64b. The process is:

1. PMU sees read request from CPU
2. Address on LMB
3. Data on LMB
4. PMU streams 64b to CPU

If the burst mode is required then extra cycles should be added, depending on the burst length:

- 1 cycle for burst 2 (BTR2)
- 3 cycles for burst 4 (BTR4)

1.3.2.2 DMU Access

The DMU Access process is:

1. DMU sees read request from CPU
2. Address on LMB
3. Data on LMB
4. Data latched inside DMI
5. DMU sends data to CPU

1.3.3 L3 Access (EBU on LMB, or Memory on FPI)

The Level 3 (L3) access time depends on the size of the external memory bus.

The assumptions in this instance are:

- An external data bus width of 32b
- No other master will require the LMB
- The delay through the EBU is zero
  (Since there are several types of EBU, the number of cycles through a specific EBU should be added to the address and data phases).

With these assumptions, access will be as follows:

1.3.3.1 PMU Access

PMU always fetches a minimum of 64b. The access process is:

1. PMU sees read request from CPU
2. Address on LMB
3. Data into L_EBU from external memory (1st 32b)
4. Data into L_EBU (2nd 32b) / Data on LMB
   (after assembling 64b, L_EBU puts data on the LMB in the same cycle)
5. PMU streams 64b to CPU

If the burst mode is required then extra cycles should be added, depending on the burst length:

- 2 cycles for burst 2 (BTR2)
- 6 cycles for burst 4 (BTR4)

1.3.3.2 DMU Access

DMU accesses are assumed to be 32b only:

1. DMU sees read request from CPU
2. Address on LMB
3. Data on LMB
4. DMU takes data from the bus
5. DMU drives data to CPU

1.3.4 L4 Access (EBU on FPI)

The Level 4 (L4) access time depends on the size of the external memory bus.

The assumptions in this instance are:

- An external data bus width of 32b
- No other master will require the LMB and/or the FPI
- The delay through the EBU is zero
  (since there are several types of EBU, the number of cycles through a specific EBU should be added to the address and data phases)

With these assumptions, access will be as follows:

1.3.4.1 PMU Access

PMU always fetches a minimum of 64b:

1. PMU sees read request from CPU
2. Address on LMB
3. Address on FPI
4. Data on FPI (1st 32b)
5. Data on FPI (2nd 32b)
6. Data on LMB

7. PMU streams 64b to CPU

If the burst mode is required then extra cycles should be added depending on the burst length:

- 2 cycles for burst 2 (BTR2)
- 6 cycles for burst 4 (BTR4)

1.3.4.2 DMU Access

DMU accesses are assumed to be 32b only:

1. DMU sees read request from CPU
2. Address on LMB
3. Address on FPI
4. Data on FPI
5. Data on LMB
6. DMU takes data from the bus
7. DMU drives data to CPU

1.4 Conclusion

This application note presents a simplified method of calculating the access time in a TC1M based system. It is up to the individual system designer to refine these calculations based on their own intimate knowledge of the system.

http://www.infineon.com

Published by Infineon Technologies AG