CEG2400 - Microcomputer Systems

Lecture 12: Buses and Memory

Philip Leong

Slides: Daeyoung Kim, Information and Communications University, Korea

http://www.arm.com/pdfs/DDI0234A_7TDMIS_R4.pdf

Introduction

- Memory interfaces
  - How the ARM processor connects to memory
- Peripheral interfaces via a bus

General

- Standardized arrangements of signals are used in a microprocessors to facilitate transfer of data
  - Called buses
  - Allow different types/sizes of memory and peripherals to be easily connected to the processor
  - Allows for customization e.g. memory configuration, devices etc

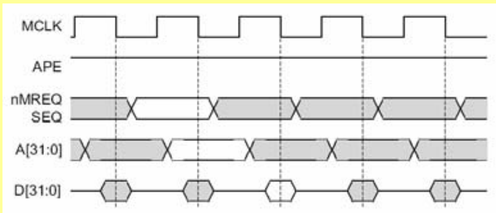
ARM memory interface

- ARM bus signals
  - A 32-bit address bus, A[31:0]
  - A 32-bit bidirectional data bus, D[31:0]
  - Control signals:
    - mreq* : memory is needed
    - seq: address is sequential
    - r*/w: direction
    - *b*/w on earlier version*; mas[1:0]: size of transfer
  - Bus timing and control signals (abe, ale, dbe, lock, bl[3:0])

The ARM memory interface

- Used for connecting an ARM processor to a memory system
- ARM bus signals
  - A 32-bit address bus, A[31:0]
  - A 32-bit bidirectional data bus, D[31:0]
  - Control signals:
    - mreq* : memory is need
    - seq: address is sequential
    - r*/w: direction
    - *b*/w on earlier version*; mas[1:0]: size of transfer
  - Bus timing and control signals (abe, ale, dbe, lock, bl[3:0])

Simple Memory Cycle



ARM7TDMI (Rev 4) Signals - 1

Name	Type	Description
A[31:0] Addresses	O	This is the 32-bit address bus. **ALE**, **ABE**, and **APE** are used to control when the address bus is valid.
ABE Address bus enable	IC	The address bus drivers are disabled when this is LOW, putting the address bus into a high impedance state. This also controls the LOCK, **MAS[1:0]**, **nRW**, **nOPC**, and **nTRANS** signals in the same way. **ABE** must be tied HIGH if there is no system requirement to disable the address drivers.
ABORT Memory abort	IC	The memory system uses this signal to tell the processor that a requested access is not allowed.
ALE Address latch enable	IC	This signal is provided for backwards compatibility with older ARM processors. For new designs, if address retiming is required, ARM Limited recommends the use of **APE**, and for **ALE** to be connected HIGH. The address bus, **LOCK**, **MAS[1:0]**, **nRW**, **nOPC**, and **nTRANS** signals are latched when this is held LOW. This allows these address signals to be held valid for the complete duration of a memory access cycle. For example, when interfacing to ROM, the address must be valid until after the data has been read.

ARM7TDMI (Rev 4) Signals - 2

APE Address pipeline enable	IC	Selects whether the address bus, **LOCK**, **MAS[1:0]**, **nRW**, **nTRANS**, and **nOPC** signals operate in pipelined (**APE** is HIGH) or depipelined mode (**APE** is LOW). Pipelined mode is particularly useful for DRAM systems, where it is desirable to provide the address to the memory as early as possible, to allow longer periods for address decoding and the generation of DRAM control signals. In this mode, the address bus does not remain valid to the end of the memory cycle. Depipelined mode can be useful for SRAM and ROM access. Here the address bus, **LOCK**, **MAS[1:0]**, **nRW**, **nTRANS**, and **nOPC** signals must be kept stable throughout the complete memory cycle. However, this does not provide optimum performance. See *Address timing* on page 3-14 for details of this timing.
BIGEND Big endian configuration	IC	Selects how the processor treats bytes in memory: • HIGH for big-endian format • LOW for little-endian format.

Pipelined & Depipelined Addresses

- Transfers are made on the rising edge of MCLK (master clock)
- APE indicates whether addresses are pipelined. If they are, they appear one cycle before the data
- Address of transfer in A, data in D

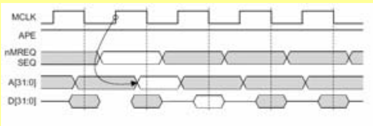


Figure 3-8 Pipelined addresses

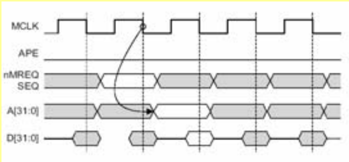


Figure 3-9 Depipelined addresses

ARM7TDMI (Rev 4) Signals - 3

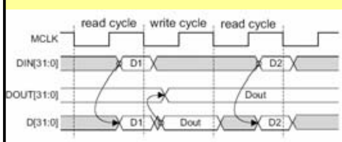
Name	Type	Description
BL[3:0] Byte latch control	IC	The values on the data bus are latched on the falling edge of **MCLK** when these signals are HIGH. For most designs these signals must be tied HIGH.
BREAKPT Breakpoint	IC	A conditional request for the processor to enter debug state is made by placing this signal HIGH. If the memory access at that time is an instruction fetch, the processor enters debug state only if the instruction reaches the execution stage of the pipeline. If the memory access is for data, the processor enters debug state after the current instruction completes execution. This allows extension of the internal breakpoints provided by the EmbeddedICE-RT logic. See *Behavior of the program counter in debug state* on page B-31 for details on the use of this signal.
BUSDIS Bus disable	O	When INTEST is selected on scan chain 0, 4, or 8 this is HIGH. It can be used to disable external logic driving onto the bidirectional data bus during scan testing. This signal changes after the falling edge of **TCK**.

ARM7TDMI (Rev 4) Signals - 4

BUSEN Data bus configuration	IC	A static configuration signal that selects whether the bidirectional data bus (**D[31:0]**) or the unidirectional data busses (**DIN[31:0]** and **DOUT[31:0]**) are used for transfer of data between the processor and memory. When **BUSEN** is LOW, **D[31:0]** is used; **DOUT[31:0]** is driven to a value of zero, and **DIN[31:0]** is ignored, and must be tied LOW. When **BUSEN** is HIGH, **DIN[31:0]** and **DOUT[31:0]** are used; **D[31:0]** is ignored and must be left unconnected. See Chapter 3 *Memory Interface* for details on the use of this signal.
CPA Coprocessor absent	IC	Placed LOW by the coprocessor if it is capable of performing the operation requested by the processor.
CPB Coprocessor busy	IC	Placed LOW by the coprocessor when it is ready to start the operation requested by the processor. It is sampled by the processor when **MCLK** goes HIGH in each cycle in which **nCPI** is LOW.

Bidirectional & Unidirectional Bus

- ARM can receive data either via D or DIN/DOUT. Configuration set by BUSEN
- Bidirectional: less wires
- Unidirectional: capable of higher speed and lower power due to reduced capacitance



Unidirectional – Inside ASIC

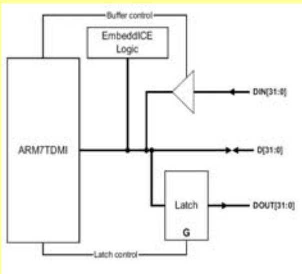


Figure 3-11 External bus arrangement

ARM7TDMI (Rev 4) Signals - 5

D[31:0] Data bus	IC O	Used for data transfers between the processor and external memory. During read cycles input data must be valid on the falling edge of **MCLK**. During write cycles output data remains valid until after the falling edge of **MCLK**. This bus is always driven except during read cycles, irrespective of the value of **BUSEN**. Consequently it must be left unconnected if using the unidirectional data buses. See Chapter 3 *Memory Interface*.
DBE Data bus enable	IC	Must be HIGH for data to appear on either the bidirectional or unidirectional data output bus. When LOW the bidirectional data bus is placed into a high impedance state and data output is prevented on the unidirectional data output bus. It can be used for test purposes or in shared bus systems.
nTRANS Not memory translate	O	When the processor is in User mode, this is LOW. It can be used either to tell the memory management system when address translation is turned on, or as an indicator of non-User mode activity. This is one of the signals controlled by **APE**, **ALE**, and **ABE**.

ARM7TDMI (Rev 4) Signals - 6

Name	Type	Description
DIN[31:0] Data input bus	IC	Unidirectional bus used to transfer instructions and data from the memory to the processor. This bus is only used when **BUSEN** is HIGH. If unused then it must be tied LOW. This bus is sampled during read cycles on the falling edge of **MCLK**.
DOUT[31:0] Data output bus	O	Unidirectional bus used to transfer data from the processor to the memory system. This bus is only used when **BUSEN** is HIGH. Otherwise it is driven to a value of zero. During write cycles the output data becomes valid while **MCLK** is LOW, and remains valid until after the falling edge of **MCLK**.
nEXEC Not executed	O	This is HIGH when the instruction in the execution unit is not being executed because, for example, it has failed its condition code check.

ARM7TDMI (Rev 4) Signals - 7

ISYNC Synchronous interrupts	IC	Set this HIGH if **nIRQ** and **nFIQ** are synchronous to the processor clock. Set it LOW for asynchronous interrupts.
LOCK Locked operation	O	When the processor is performing a locked memory access this is HIGH. This is used to prevent the memory controller allowing another device to access the memory. It is active only during the data swap (SWP) instruction. This is one of the signals controlled by **APE**, **ALE** and **ABE**.
MAS[1:0] Memory access size	O	Used to indicate to the memory system the size of data transfer (byte, halfword or word) required for both read and write cycles, become valid before the falling edge of **MCLK** and remain valid until the rising edge of **MCLK** during the memory cycle. The binary values 00, 01, and 10 represent byte, halfword, and word respectively (11 is reserved). This is one of the signals controlled by **APE**, **ALE**, and **ABE**.
MCLK Memory clock input	IC	This is the main clock for all memory accesses and processor operations. The clock speed can be reduced to allow access to slow peripherals or memory. Alternatively, the **nWAIT** can be used with a free-running **MCLK** to achieve the same effect.

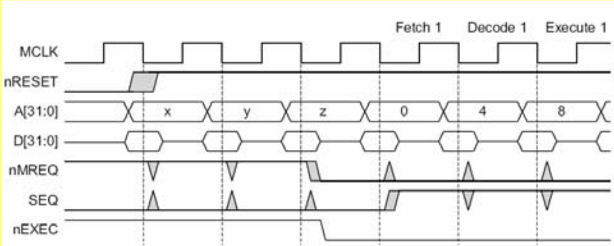
ARM7TDMI (Rev 4) Signals - 8

Name	Type	Description
nCPI Not coprocessor instruction	O	LOW when a coprocessor instruction is processed. The processor then waits for a response from the coprocessor on the **CPA** and **CPB** lines. If **CPA** is HIGH when **MCLK** rises after a request has been initiated by the processor, then the coprocessor handshake is aborted, and the processor enters the undefined instruction trap. If **CPA** is LOW at this time, then the processor enters a busy-wait period until **CPB** goes LOW before completing the coprocessor handshake.
nFIQ Not fast interrupt request	IC	Taking this LOW causes the processor to be interrupted if the appropriate enable in the processor is active. The signal is level-sensitive and must be held LOW until a suitable response is received from the processor. **nFIQ** can be synchronous or asynchronous to **MCLK**, depending on the state of **ISYNC**.
nIRQ Not interrupt request	IC	As **nFIQ**, but with lower priority. Can be taken LOW to interrupt the processor when the appropriate enable is active. **nIRQ** can be synchronous or asynchronous, depending on the state of **ISYNC**.
nM[4:0] Not processor mode	O	These are the inverse of the internal status bits indicating the current processor mode.

ARM7TDMI (Rev 4) Signals - 9

nMREQ Not memory request	O	When the processor requires memory access during the following cycle this is LOW.
nOPC Not op-code fetch	O	When the processor is fetching an instruction from memory this is LOW. This is one of the signals controlled by **APE**, **ALE**, and **ABE**.
nRESET Not reset	IC	Used to start the processor from a known address. A LOW level causes the instruction being executed to terminate abnormally. This signal must be held LOW for at least two clock cycles, with **nWAIT** held HIGH. When LOW the processor performs internal cycles with the address incrementing from the point where reset was activated. The address overflows to zero if **nRESET** is held beyond the maximum address limit. When HIGH for at least one clock cycle, the processor restarts from address 0.
nRW Not read, write	O	When the processor is performing a read cycle, this is LOW. This is one of the signals controlled by **APE**, **ALE**, and **ABE**.

Reset Timing



ARM7TDMI (Rev 4) Signals - 10

nWAIT Not wait	IC	When LOW the processor extends an access over a number of cycles of **MCLK**, which is useful for accessing slow memory or peripherals. Internally, **nWAIT** is logically ANDed with **MCLK** and must only change when **MCLK** is LOW. If **nWAIT** is not used it must be tied HIGH.
SEQ Sequential address	O	When the address of the next memory cycle is closely related to that of the last memory access, this is HIGH. In ARM state the new address can be for the same word or the next. In THUMB state, the same halfword or the next. It can be used, in combination with the low-order address lines, to indicate that the next cycle can use a fast memory mode (for example DRAM page mode) or to bypass the address translation system.
VDD Power supply	P	Provide power to the device.
VSS Ground	P	These connections are the ground reference for all signals.

Memory Cycles

The ARM7TDMI-S processor bus interface can perform four different types of memory cycle. These are indicated by the state of the **TRANS[1:0]** signals. Memory cycle types are encoded on the **TRANS[1:0]** signals as shown in Table 3-1.

Table 3-1 Cycle types

TRANS[1:0]	Cycle type	Description
00	I cycle	Internal cycle
01	C cycle	Coprocessor register transfer cycle
10	N cycle	Nonsequential cycle
11	S cycle	Sequential cycle

A memory controller for the ARM7TDMI-S processor commits to a memory access only on an N cycle or an S cycle.

Memory cycles

Nonsequential cycle

During this cycle, the ARM7TDMI-S core requests a transfer to, or from an address which is unrelated to the address used in the preceding cycle.

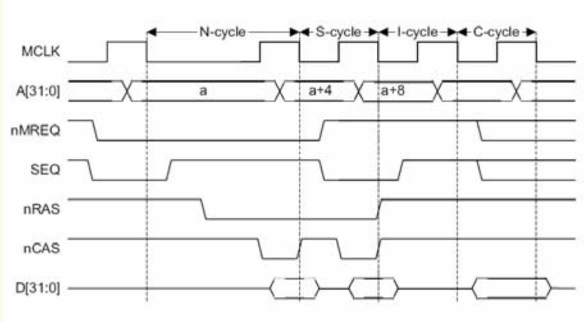
Sequential cycle During this cycle, the ARM7TDMI-S core requests a transfer to or from an address that is either one word or one halfword greater than the address used in the preceding cycle.

Internal cycle During this cycle, the ARM7TDMI-S core does not require a transfer because it is performing an internal function and no useful prefetching can be performed at the same time.

Coprocessor register transfer cycle

During this cycle, the ARM7TDMI-S core uses the data bus to communicate with a coprocessor but does not require any action by the memory system.

Memory Cycle Timing



nWAIT

Used to make a memory cycle longer so memory device has more time to respond

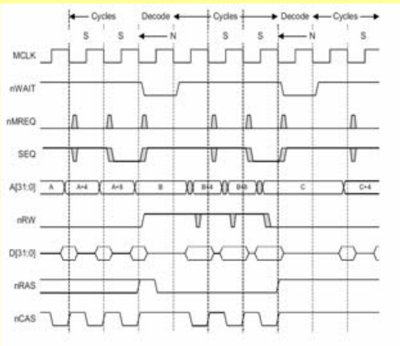
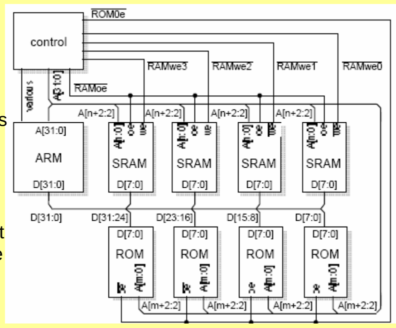


Figure 3-21 Typical system timing

Basic memory system

- Simplest for ROM and static RAM (SRAM)
- Address bus connected together
- Each SRAM/ROM responsible for 8-bits of 32-bit data bus
- Oe and we used to enable output and write of chip so multiple chips do not try to drive the same signal

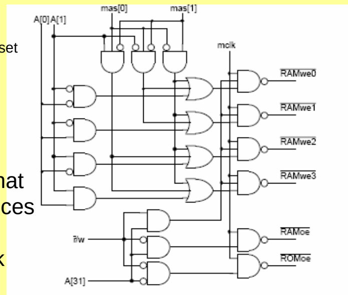


Control logic

- Decides when to activate ROM and when to activate RAM
  - This logic determines system memory map i.e. which address corresponds to which memory device
- Controls byte write enables during a write operation
  - For word write, all write enables must be asserted. For say byte write to address 0x8000003, only assert SRAM3
- Ensures data ready before processor continues
  - In the following design, assume memory is fast enough. Otherwise, need to use nWAIT

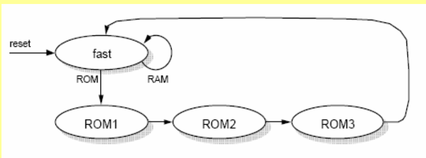
Simple ARM memory system control logic

- ROM or RAM
  - Memory map
    - Start at zero after reset (ROM)
    - A31=0 (ROM)
    - A31=1 (RAM)
- Run mclk slowly enough to ensure that all the memory devices can be accessed within a single clock cycle



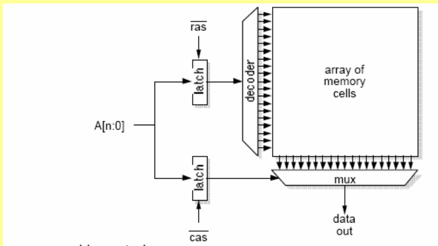
ROM wait control state transition diagram

- If we speed up the mclk, it will stop working when the slowest path fails (ROM access)
  - To achieve higher performance, clock is tuned to RAM access time
    - Wait states to give more access time to ROM (assume four clocks)



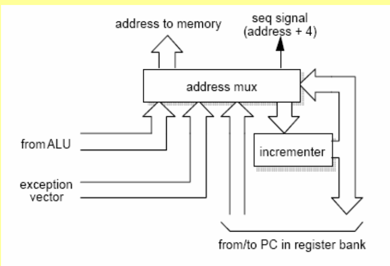
DRAM memory organization

- Dynamic memory made from single transistor memory elements
- The row and address signals are multiplexed to reduce pins. Hence we need to present the row part of the address, do a ras, present column, do a cas to address a single cell

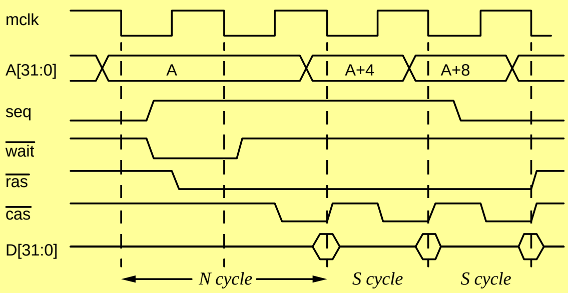


ras: row address strobe ;
cas: column address strobe

ARM address register structure



DRAM timing illustration



Memory Quiz

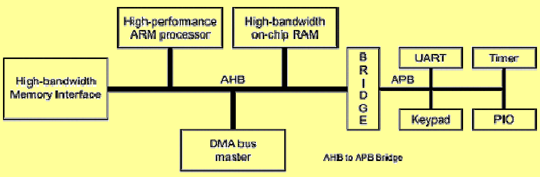
- For the ARM memory system control logic slide, explain what is required for nRAMwe3 to be asserted
- What is the difference between a pipelined and non-pipelined memory cycle? What signal controls this function?
- Why is nWAIT needed?

Memory

- Must be a write cycle with A[31] high AND (A[0]A[1] + A[1].mas[0]./mas[1] + /mas[0].mas[1]) AND MCLK high
  - A[31] is upper half of address space
  - A[0]A[1] means address is 11 in bottom 2 bits
  - A[1].mas[0]./mas[1] means halfword access and A[1] set
  - /mas[0].mas[1] means word access
  - In words, for a write cycle to be made to SRAM3, it needs to be in the upper half of the address space i.e. address >= 0x8000000 and either (bottom bits are “11” or halfword access and A[1]=1 or word access). See description for mas signal in earlier slide
- See notes for other 2 questions

AMBA - 1

- Advanced Microcontroller Bus Architecture
  - Standard of on-chip communication between different macrocells for high performance embedded system design
  - Hierarchical Bus architecture



AMBA - 2

- AMBA buses
  - AHB(Advanced High Performance Bus)
    - Connect between high-performance system modules
  - ASB(Advanced System Bus)
    - Subset of AHB
  - APB(Advanced Peripheral Bus)
    - Simple interface for low-performance peripherals

AMBA - 3

- AMBA buses (cont'd)

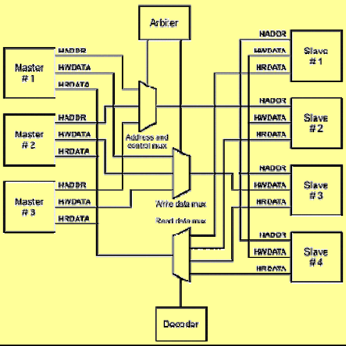
AHB	ASB	APB
- burst transfers - split transactions - single-cycle bus master handover - single-clock edge operation - wider data bus configurations (64/128 bits) - multiple bus masters (up to 16) - pipelined operation	- burst transfers - pipelined operation - multiple bus masters	- low power - latched address and control - simple interface - suitable for many peripherals

AMBA - 4

- AMBA AHB component
  - Master
    - Initiate read and write operations by providing an address and control information. Only one bus master is allowed to actively use the bus at any one time.
  - Slave
    - Responds to a read or write operation within a given address-space range. The bus slave signals back to the active master the success, failure or waiting of the data transfer.
  - Arbiter
    - Ensures that only one bus master at a time is allowed to initiate data transfers. Can use the priority
  - Decoder
    - Decode the address of each transfer and provide a select signal for the slave that is involved in the transfer.

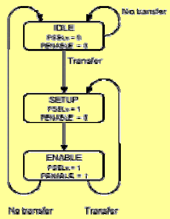
AMBA - 5

- AMBA AHB component (cont'd)



AMBA - 6

- AMBA APB
  - APB bridge: only master in APB. Act as slave in AHB
  - APB slave: peripherals
  - Simple protocol



- Processor core
  - Master in AHB
  - Connect through the memory interface of core