GLOBAL EDGE
Intelligence Of Things

ARM - Architecture

Vishwanath B

Introduction : ARM Ltd

- Founded in November 1990
  - Spun out of Acorn Computers
- Designs the ARM range of RISC processor cores
- Licenses ARM core designs to semiconductor partners who fabricate and sell to their customers.
  - ARM does not fabricate silicon itself
- Also develop technologies to assist with the design-in of the ARM architecture
  - Software tools, boards, debug hardware, application software, bus architectures, peripherals etc

ARM Partnership Model

GLOBAL EDGE

Introduction : ARM7TDMI-S

- The ARM7TDMI-S processor is a member of the Advanced RISC machine family of general purpose 32-bit microprocessor

- What does mean ARM7TDMI-S ?

  ARM7 - 32-bit Advanced RISC Machine

  T - Thumb architecture extension
    - Two separate instruction sets, 32-bit ARM instructions and 16-bit Thumb instructions

  D - Debug extension

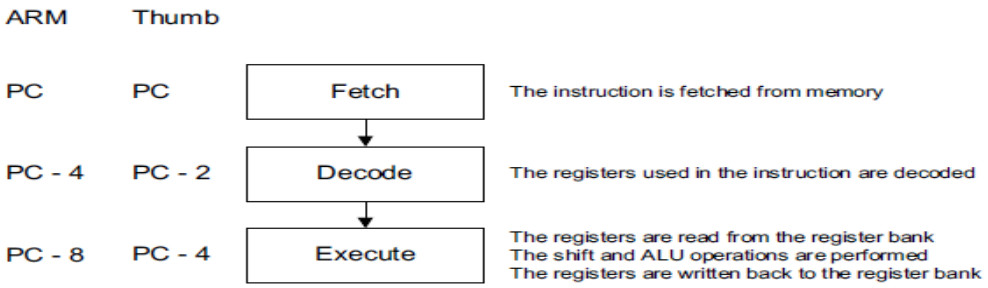
  M - Enhanced multiplier

  I - Embedded ICE macrocell extension

  S - Synthesizable core

Introduction : ARM7TDMI-S

- ARM7TDMI-S processor is a member of the ARM family of general-purpose 32-bit microprocessors.
- The ARM family offers high performance for very low-power consumption and gate count.
- The ARM architecture is based on *Reduced Instruction Set Computer (RISC)* principles.
- A three-stage pipeline is used, so instructions are executed in three stages:



Introduction : Memory access

- The ARM7TDMI-S processor has a Von- Neumann architecture, with a single 32-bit data bus carrying both instructions and data.
- Only load, store, and swap instructions can access data from memory.
- Data can be 8-bit bytes, 16-bit half words, or 32-bit words. Words must be aligned to 4-byte boundaries. Half words must be aligned to 2-byte boundaries.
- The ARM7TDMI-S processor has four basic types of memory cycle:
  - Internal cycle
  - Non sequential cycle
  - Sequential cycle
  - Coprocessor register transfer cycle.

Introduction : Instruction compression

- The ARM7TDMI-S processor has two instruction sets:
  - 32-bit ARM instruction set
  - 16-bit Thumb instruction set.
- The ARM7TDMI-S processor is an implementation of the ARM architecture v4T.
- Microprocessor architectures traditionally had the same width for instructions and data.
- Thumb implements a 16-bit instruction set on a 32-bit architecture to provide:
  - higher performance than a 16-bit architecture
  - higher code density than a 32-bit architecture.

Arm Instruction Set

- All instructions are 32 bits long.
- Most instructions are executed in one single cycle.
- Every instructions can be conditionally executed.
- A load/store architecture
  - Data processing instructions act only on registers
    - Three operand format
    - Combined ALU and shifter for high speed bit manipulation
  - Specific memory access instructions with powerful auto-indexing addressing modes
  - 32 bit ,16 bit and 8 bit data types
  - Flexible multiple register load and store instructions

Thumb instruction set

- Thumb instructions are each 16 bits long, and have a corresponding 32-bit ARM instruction that has the same effect on the processor model.

- Thumb instructions operate with the standard ARM register configuration, allowing excellent interoperability between ARM and Thumb states.

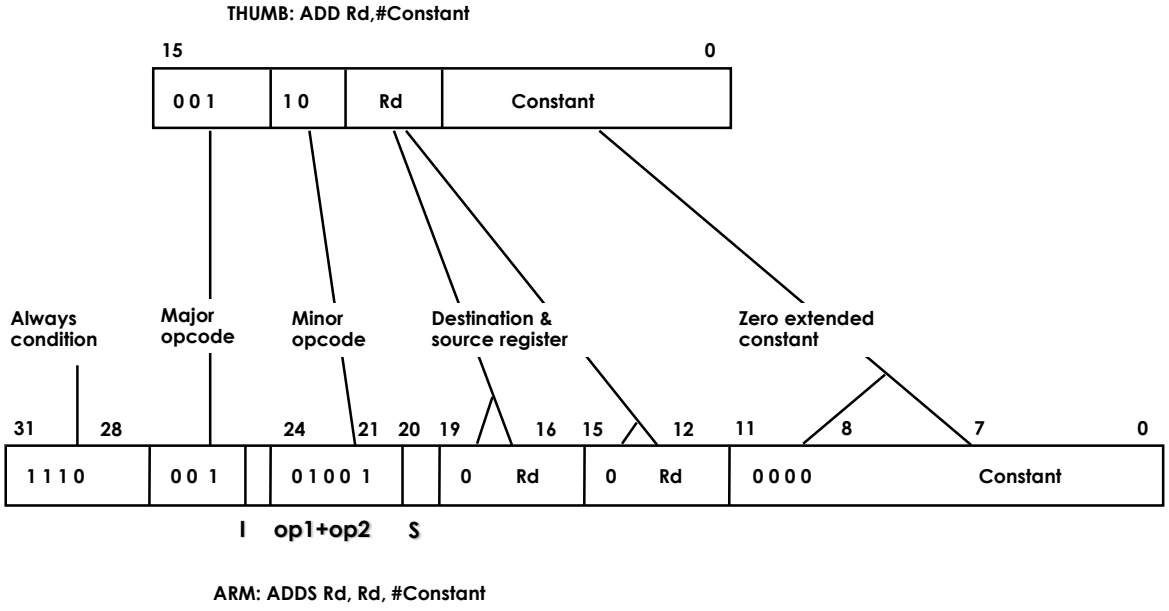
- Thumb has all the advantages of a 32-bit core:
  - 32-bit address space
  - 32-bit registers
  - 32-bit shifter and *Arithmetic Logic Unit (ALU)*
  - 32-bit memory transfer.

- Thumb therefore offers a long branch range, powerful arithmetic operations, and a large address space.

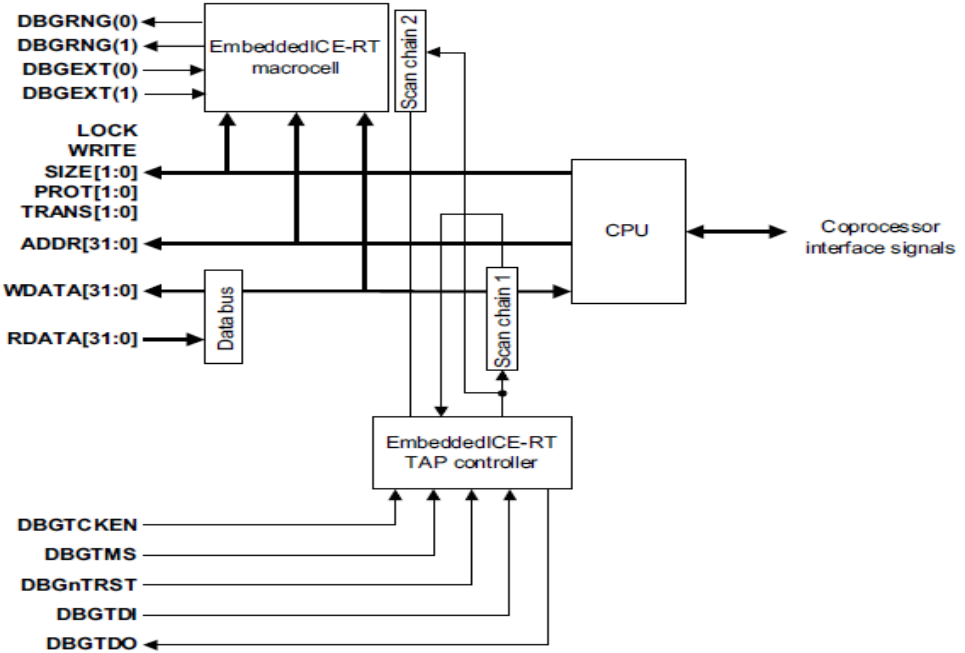
Thumb instruction set

- Thumb code is typically 65% of the size of the ARM code.
- Thumb code provides 160% of the performance of ARM code when running on a processor connected to a 16-bit memory system.
- The availability of both 16-bit Thumb and 32-bit ARM instruction sets gives designers the flexibility to emphasize performance, or code size.
  - Example : Fast interrupts and DSP algorithms can be coded using the full ARM instruction set and linked with Thumb code.
- Almost every Thumb instructions have an ARM instructions equivalent:
  - ADD Rd, #Offset8 <> ADDS Rd, Rd, #Offset8

Thumb Instruction Set Decompression

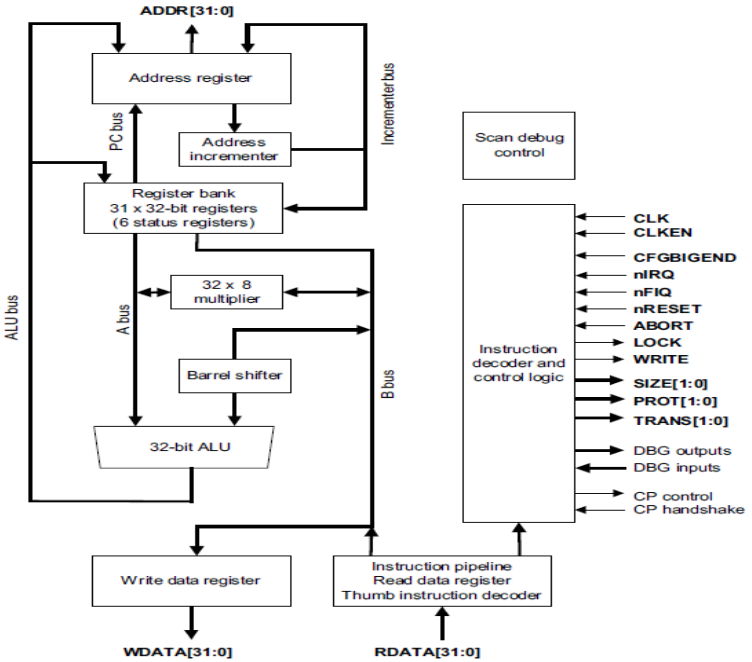


ARM7TDMI-S block diagram

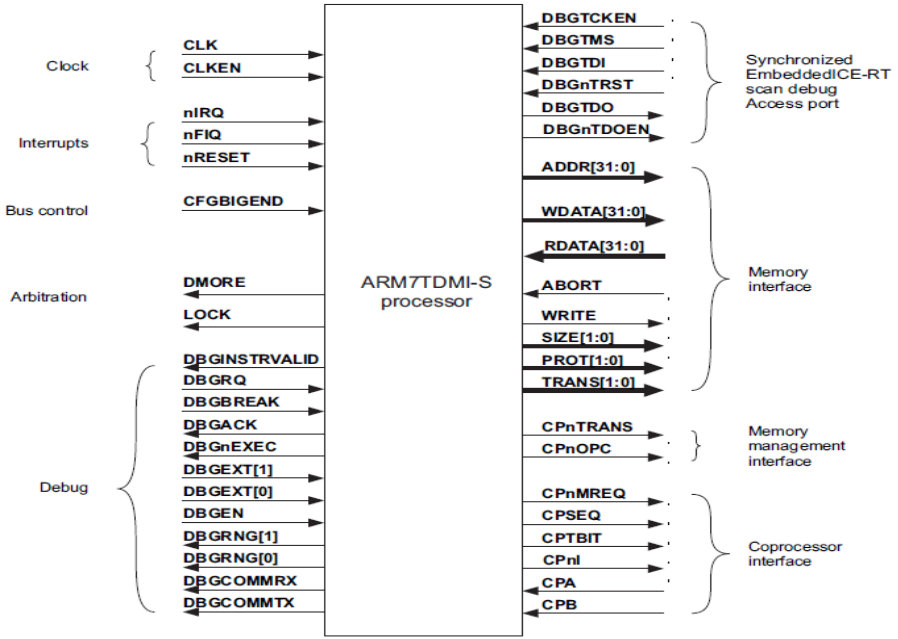


ARM7TDMI Block Diagram

- Von Neumann Architecture
- 3-stage pipeline
  - fetch, decode, execute
- 32-bit Data Bus
- 32-bit Address Bus
- 37 : 32-bit registers
- 32-bit ARM instruction set
- 16-bit THUMB instruction set
- 32x8 Multiplier
- Barrel Shifter



ARM7TDMI-S functional diagram



Programmer's Model

- The ARM7TDMI-S processor has two operating states:
  - ARM state 32-bit, word-aligned ARM instructions are executed in this state.
  - Thumb state 16-bit, halfword-aligned Thumb instructions.
- In Thumb state, the *Program Counter (PC) uses bit 1 to select between alternate* halfwords.
- Transition between ARM and Thumb states does not affect the processor mode or the register contents.
- You can switch between ARM state and Thumb state using the BX instruction.
- All exception handling is performed in ARM state.
- If an exception occurs in Thumb state, the processor reverts to ARM state. The transition back to Thumb state occurs automatically on return.

Memory formats

- The ARM7TDMI-S processor views memory as a linear collection of bytes numbered in ascending order from zero

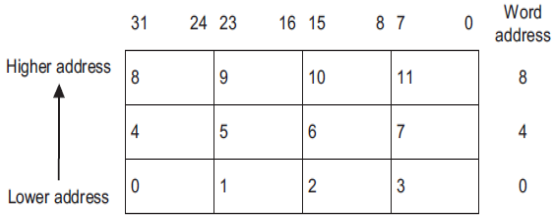
- The ARM7TDMI-S processor can treat words in memory as being stored in one of:
  - *Big-endian format*
  - *Little-endian format.*

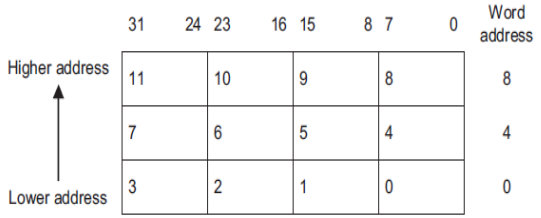
Big-endian format

➢ Byte 0 of the memory system connects to data lines 31 to 24

Little-endian format

➢ Byte 0 of the memory system connects to data lines 7 to 0.





Instruction length & Data types

- Instructions are either:
  - 32 bits long (in ARM state)
  - 16 bits long (in Thumb state).

- The ARM7TDMI-S processor supports the following data types:
  - word (32-bit)
  - halfword (16-bit)
  - byte (8-bit).

- You must align these as follows:
  - word quantities must be aligned to four-byte boundaries
  - halfword quantities must be aligned to two-byte boundaries
  - byte quantities can be placed on any byte boundary.

Operating modes

- The ARM7TDMI-S processor has seven operating modes:
  - User mode is the usual ARM program execution state, and is used for executing most application programs.
  - Fast interrupt (FIQ) mode supports a data transfer or channel process.
  - Interrupt (IRQ) mode is used for general-purpose interrupt handling.
  - Supervisor mode is a protected mode for the operating system.
  - Abort mode is entered after a data or instruction prefetch abort.
  - System mode is a privileged user mode for the operating system.
  - Undefined mode is entered when an undefined instruction is executed.
- Modes other than User mode are collectively known as privileged modes.
- Privileged modes are used to service interrupts, exceptions, or access protected resources.

Registers

- The ARM7TDMI-S processor has a total of 37 registers:
  - 31 general-purpose 32-bit registers
  - 6 status registers.
- The processor state and operating mode determine which registers are available to the programmer.

- The ARM state register set :
  - The ARM state register set contains 16 directly-accessible registers, r0 to r15 & CPSR(Current Program Status Register)

  - Registers r0 to r13 are general-purpose registers used to hold either data or address values.

  - Register 14 is used as the subroutine Link Register (LR).r14 receives a copy of r15 when a Branch with Link (BL)instruction is executed.

  - Register 15 holds the Program Counter (PC).
    - In ARM state, bits [1:0] of r15 are zero. Bits [31:2] contain the PC.
    - In Thumb state, bit [0] is zero. Bits [31:1] contain the PC.

- Saved Program Status Register (SPSR) is Accessible In privileged modes . This contains CPSR bits saved as a result of the exception that caused entry to the current mode

ARM state general registers & program counter

System and User	FIQ	Supervisor	Abort	IRQ	Undefined
r0	r0	r0	r0	r0	r0
r1	r1	r1	r1	r1	r1
r2	r2	r2	r2	r2	r2
r3	r3	r3	r3	r3	r3
r4	r4	r4	r4	r4	r4
r5	r5	r5	r5	r5	r5
r6	r6	r6	r6	r6	r6
r7	r7	r7	r7	r7	r7
r8	r8_fiq	r8	r8	r8	r8
r9	r9_fiq	r9	r9	r9	r9
r10	r10_fiq	r10	r10	r10	r10
r11	r11_fiq	r11	r11	r11	r11
r12	r12_fiq	r12	r12	r12	r12
r13	r13_fiq	r13_svc	r13_abt	r13_irq	r13_und
r14	r14_fiq	r14_svc	r14_abt	r14_irq	r14_und
r15 (PC)	r15 (PC)	r15 (PC)	r15 (PC)	r15 (PC)	r15 (PC)

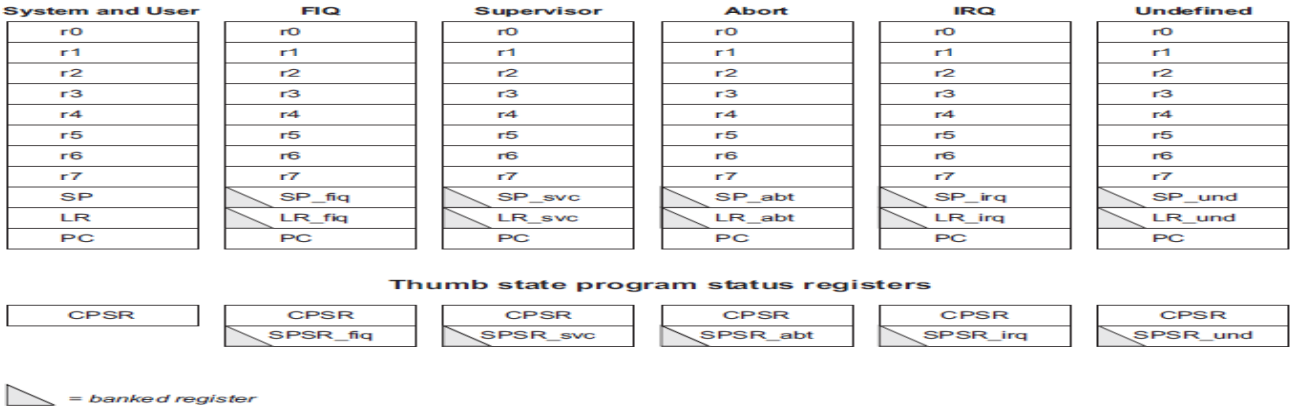
ARM state program status registers

CPSR	CPSR	CPSR	CPSR	CPSR	CPSR
	SPSR_fiq	SPSR_svc	SPSR_abt	SPSR_irq	SPSR_und

= *banked register*

Thumb state register set

- The Thumb state register set is a subset of the ARM state set. The programmer has direct access to: eight general registers, r0–r7, the PC, a Stack Pointer (SP),a Link Register (LR) , the CPSR.
- There are banked SPs, LRs, and SPSRs for each privileged mode.



Relationship between ARM state & Thumb state Registers

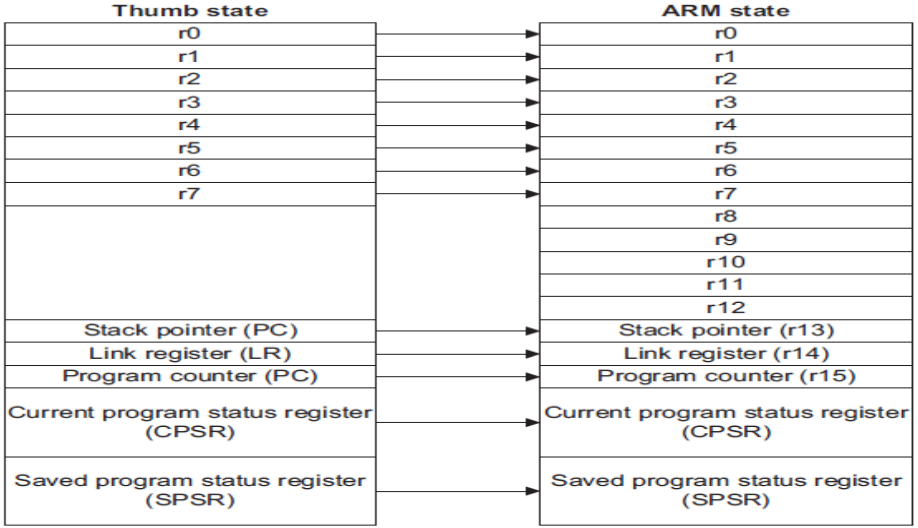


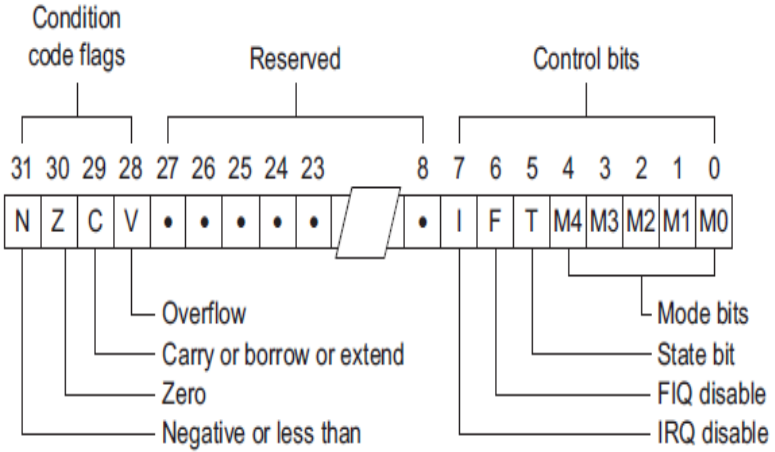
Figure 2-5 Mapping of Thumb state registers onto ARM state registers

Note

Registers r0–r7 are known as the low registers. Registers r8–r15 are known as the high registers.

Program Status Registers

The ARM7TDMI-S core contains a CPSR and five SPSRs for exception handlers to
use



Exceptions

- Exception are events that occur in a system that are not expected or are not a part of normal system operation. When the system handles these exceptional conditions improperly, it can lead to failures and system crashes.

Event	Exception	Priority [1]	Return address	Status	Mode	FIQ	IRQ	Vector [2]	Prefered return instruction
Reset input de-asserted	Reset	1	Not available	Not available	Supervisor	Disabled	Disabled	Base+0	Not available
Reading from or writing to invalid address	Data Access Memory Abort (Data Abort)	2	R14_abt=PC+8 [4]	SPSR_abt=CPSR	Abort	Unchanged	Disabled	Base+16	SUBS PC,R14_abt,#8 [8]
FIQ input asserted	Fast Interrupt (FIQ)	3	R14_fiq=PC+4 [5]	SPSR_fiq=CPSR	FIQ	Disabled	Disabled	Base+28 [7]	SUBS PC,R14_fiq,#4
IRQ input asserted	Normal Interrupt (IRQ)	4	R14_irq=PC+4 [5]	SPSR_irq=CPSR	IRQ	Unchanged	Disabled	Base+24	SUBS PC,R14_irq,#4
Executing BKPT [3] or instruction at invalid address	Instruction Fetch Memory Abort (Prefetch Abort)	5	R14_abt=PC+4 [6]	SPSR_abt=CPSR	Abort	Unchanged	Disabled	Base+12	SUBS PC,R14_abt,#4
Executing SWI instruction	Software Interrupt (SWI)	6	ARM state: R14_svc=PC+4 Thumb state: R14_svc=PC+2 [6]	SPSR_svc=CPSR	Supervisor	Unchanged	Disabled	Base+8	MOVS PC,R14_svc
Executing undefined instruction code	Undefined Instruction	6	ARM state: R14_und=PC+4 Thumb state: R14_und=PC+2 [6]	SPSR_und=CPSR	Undefined	Unchanged	Disabled	Base+4	MOVS PC,R14_und

Interrupt latencies

- *Maximum interrupt latencies*

  When FIQs are enabled, the worst-case latency for FIQ comprises a combination of:

  - Tsyncmax, the longest time the request can take to pass through the synchronizer. Tsyncmax is two processor cycles.
  - Tldm, the time for the longest instruction to complete. (The longest instruction is an LDM that loads all the registers including the PC.) Tldm is 20 cycles in a zero wait state system.
  - Texc, the time for the Data Abort entry. Texc is three cycles.
  - Tfiq, the time for FIQ entry. Tfiq is two cycles.

- The total latency is therefore 27 processor cycles, slightly less than 0.7 microseconds in a system that uses a continuous 40MHz processor clock.

- *Minimum interrupt latencies*
  - The minimum latency for FIQ or IRQ is the shortest time the request can take through the synchronizer, Tsyncmin plus Tfiq (four processor cycles).

Reset

- When the nRESET signal goes LOW, the ARM7TDMI-S processor abandons the executing instruction.

- When nRESET goes HIGH again the ARM7TDMI-S processor:
  1. Forces M[4:0] to b10011 (Supervisor mode).
  2. Sets the I and F bits in the CPSR.
  3. Clears the CPSR T bit.
  4. Forces the PC to fetch the next instruction from address 0x00.
  5. Reverts to ARM state and resumes execution.

- After reset, all register values except the PC and CPSR are indeterminate.

Large enough to Deliver, Small enough to Care

Global Village
IT SEZ
Bangalore

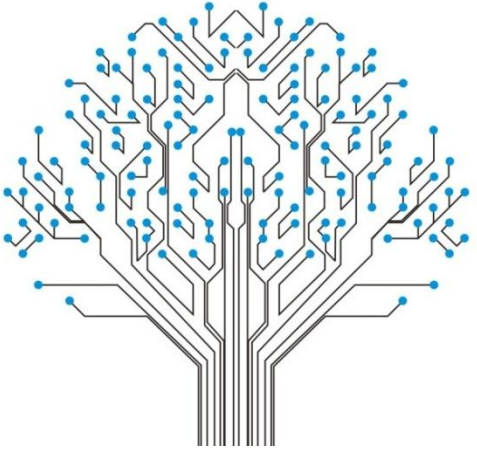
South Main Street
Milpitas
California

Raheja Mindspace
IT Park
Hyderabad

www.globaledgesoft.com

GLOBAL EDGE
Intelligence Of Things

Thank you



Fairness

Learning

Responsibility

Innovation

Respect