# Org: ARM

Alex Chi

*Update: April 23, 2020*

## Contents

**1 Registers** **2**

1.1 R14 . . . 2

1.2 R15 (Program Counter) . . . 2

1.3 PSR (Program Statue Registers) . . . 2

1.4 PRIMASK, FAULTMASK, BASEPRI . . . 2

1.5 Control . . . 3

1.6 Operation Mode in Cortex M3/M4 . . . 3

1.7 Exception . . . 3

1.8 Interrupt Vector Table . . . 3

1.9 Reset Sequence . . . 3

**2 Memory System** **4**

2.1 Connection . . . 4

2.2 Bit-Band operations . . . 4

2.3 Data Alignment and Unaligned Transfer . . . 4

2.4 Endian . . . 4

# 1 Registers

- thumb R0-R7
- thumb-2 R8-R12
- R13 banked two stack pointers
- control register: privileged mode, stack pointer
- full descending stack management

## 1.1 R14

- bl (branch and link): save next instruction address to R14
- RISC-V: jal (jump and link)

## 1.2 R15 (Program Counter)

- b0 always 0 (2/4 byte instruction)
- can move to PC (lsb = 1, thumb; lsb = 0, arm, fault in Cortex-M3)
- points to next instruction to this one

## 1.3 PSR (Program Statue Registers)

- subdivided into 3: Application, Interrupt, Execution
- APSR, IPSR, EPSR
- APSR: NZCVQ (Negative, Zero, Carry, Overflow, Sticky saturation)
- EPSR: if then
- IPSR: exception number
- xPSR: access together

## 1.4 PRIMASK, FAULTMASK, BASEPRI

- pri, fault = 1-bit
- base 9-bit, specify priority

## 1.5 Control

- control[0], privileged (0) in thread mode
- control[1], default stack (0) / alternate

## 1.6 Operation Mode in Cortex M3/M4

- two modes and two privilege levels
- privileged thread / user thread / privileged handler
- push to PSP when interrupt
- tell CPU which stack to pop when interrupt exit

## 1.7 Exception

- 1 reset (priority -3)
- 2 NMI (-2)
- 3 hard fault (-1)
- 4 mem manage
- 5 bus fault
- 6 usage fault
- etc.

## 1.8 Interrupt Vector Table

- INT 01 = 0000004 (no INT 00)
- LSB of all vectors in table must set to 1 (thumb mode)
- 0x00: starting value of MSP

## 1.9 Reset Sequence

- read R13 from 0x0
- jump to reset interrupt handler

# 2 Memory System

- MMIO
- 0.5GB
- 0.5GB SRAM
- 0.5GB Peripherals
- 1GB RAM, 1GB Devices
- System / Vendor specific

## 2.1 Connection

- I-code/D-code Bus Mux
- System bus - SRAM, peripheral bus bridge
- queue in processor

## 2.2 Bit-Band operations

- bit-band region and bit-band alias
- mutual exclusion read/write, no interrupt during bit read/write

## 2.3 Data Alignment and Unaligned Transfer

- support unaligned transfer
- converted into multiple aligned transfers

## 2.4 Endian

- big and little endian (recommended)