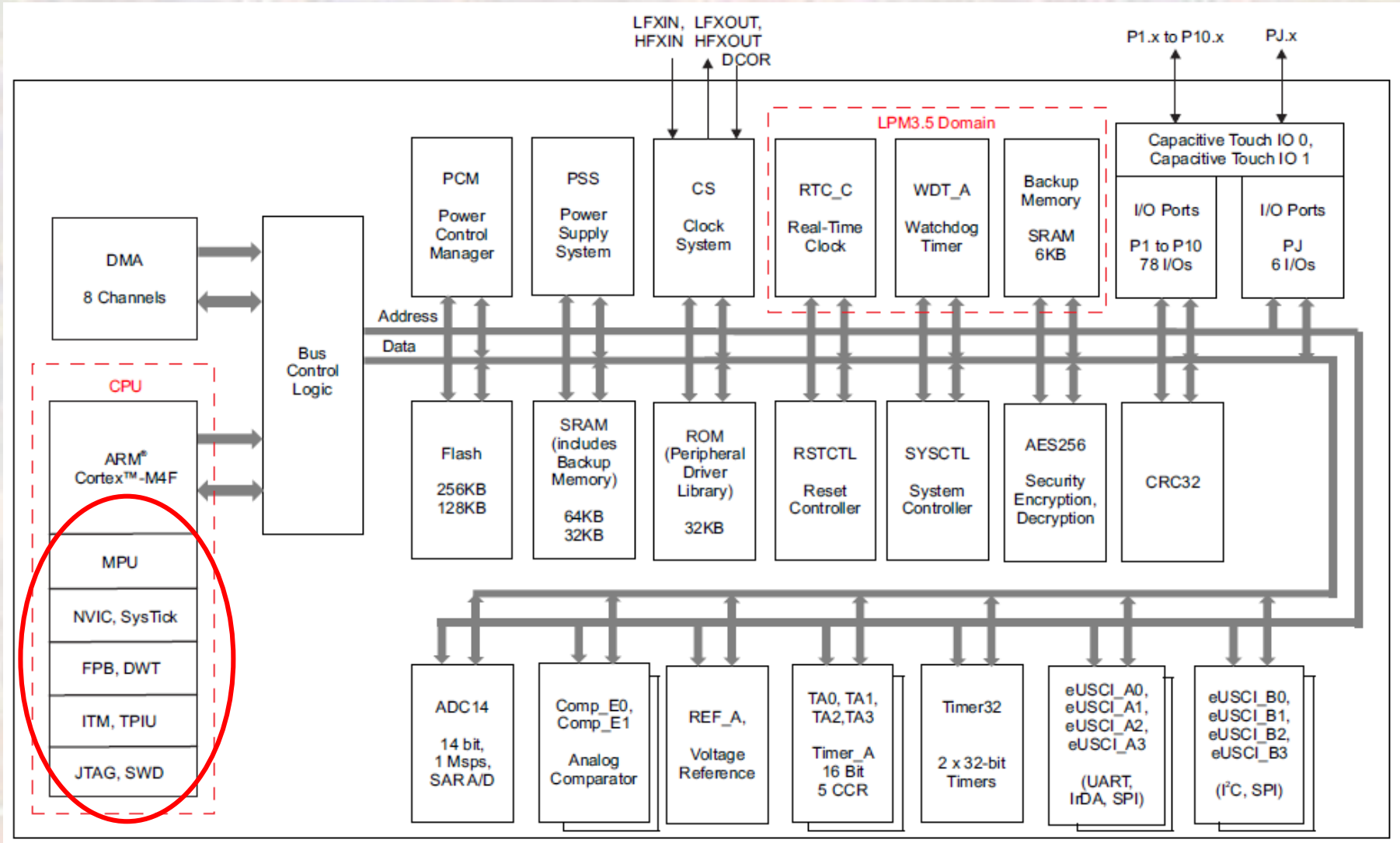
ARM M4 Peripherals

Last updated 7/8/19

ARM M4 Peripherals



ARM M4 Peripherals

- Core Peripherals
  - SysTick
  - System Control Block
  - Memory Protection Unit
  - Floating Point Unit
  - Debug Peripherals
  - Nested Vector Interrupt Controller(in interrupts notes)

ARM M4 Peripherals

- Systick
  - Integrated system timer (counter)
  - 24-bit - clear-on-write (reloads)
  - Decrementing
  - Wrap-on-zero

ARM M4 Peripherals

- Systick
  - Applications
    - An RTOS tick timer that fires at a programmable rate (for example, 100 Hz) and invokes a SysTick routine.
    - A high-speed alarm timer using the system clock.
    - A variable rate alarm or signal timer—the duration is range-dependent on the reference clock used and the dynamic range of the counter.
    - A simple counter used to measure time to completion and time used.
    - An internal clock source control based on missing/meeting durations.
    - The COUNT bit in the STCSR control and status register can be used to determine if an action completed within a set duration, as part of a dynamic clock management control loop

ARM M4 Peripherals

- Systick

1. Program the value in the STRVR register.
2. Clear the STCVR register by writing to it with any value.
3. Configure the STCSR register for the required operation.

Table 2-53. SYSTICK Registers

Offset	Acronym	Register Name	Type	Reset	Section
10h	STCSR	SysTick Control and Status Register	read-write	00000004h	Section 2.4.4.1
14h	STRVR	SysTick Reload Value Register	read-write	Undefined	Section 2.4.4.2
18h	STCVR	SysTick Current Value Register	read-write	Undefined	Section 2.4.4.3
1Ch	STCR	SysTick Calibration Value Register	read-only	Undefined	Section 2.4.4.4

SysTick -> CTRL
LOAD
VAL
CALIB

ARM M4 Peripherals

- Systick – Control Register

SysTick -> CTRL

Figure 2-47. STCSR Register

31	30	29	28	27	26	25	24
RESERVED							
R/W-0h							
23	22	21	20	19	18	17	16
RESERVED							COUNTFLAG
R/W-0h							R-0h
15	14	13	12	11	10	9	8
RESERVED							
R/W-0h							
7	6	5	4	3	2	1	0
RESERVED					CLKSOURCE	TICKINT	ENABLE
R/W-0h					R-1h	R/W-0h	R/W-0h

Table 2-54. STCSR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-17	RESERVED	R/W	0h	
16	COUNTFLAG	R	0h	Returns 1 if timer counted to 0 since last time this was read. Clears on read by application of any part of the SysTick Control and Status Register. If read by the debugger using the DAP, this bit is cleared on read-only if the MasterType bit in the AHB-AP Control Register is set to 0. Otherwise, the COUNTFLAG bit is not changed by the debugger read.
15-3	RESERVED	R/W	0h	
2	CLKSOURCE	R	1h	Clock source. 0b = Not applicable 1b = Core clock
1	TICKINT	R/W	0h	
0	ENABLE	R/W	0h	Enable SysTick counter 0b (R/W) = Counter disabled

ARM M4 Peripherals

- Systick – Reload Value Register

SysTick -> LOAD

Figure 2-48. STRVR Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								RELOAD																							
R/W								R/W																							

Table 2-55. STRVR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	RESERVED	R/W	Undefined	
23-0	RELOAD	R/W	Undefined	Value to load into the SysTick Current Value Register when the counter reaches 0.

ARM M4 Peripherals

- Systick – Current Value Register

SysTick -> VAL

Figure 2-49. STCVR Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								CURRENT																							
R/W								R/W																							

Table 2-56. STCVR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	RESERVED	R/W	Undefined	
23-0	CURRENT	R/W	Undefined	Current value at the time the register is accessed. No read-modify-write protection is provided, so change with care. Writing to it with any value clears the register to 0. Clearing this register also clears the COUNTFLAG bit of the SysTick Control and Status Register.

Note: Writing anything to this register clears the count

ARM M4 Peripherals

- Systick – Calibration Register

SysTick -> CALIB

Figure 2-50. STCR Register

31	30	29	28	27	26	25	24
NOREF	SKEW	RESERVED					
R-	R-	R-					
23	22	21	20	19	18	17	16
TENMS							
R-							
15	14	13	12	11	10	9	8
TENMS							
R-							
7	6	5	4	3	2	1	0
TENMS							
R-							

Table 2-57. STCR Register Field Descriptions

Bit	Field	Type	Reset	Description
31	NOREF	R	Undefined	Reads as one. Indicates that no separate reference clock is provided.
30	SKEW	R	Undefined	Reads as one. The calibration value is not exactly 10ms because of clock frequency. This could affect its suitability as a software real time clock.
29-24	RESERVED	R	Undefined	
23-0	TENMS	R	Undefined	Reads as zero. Indicates calibration value is not known.

ARM M4 Peripherals

- SysTick Example

```
int Delay_48MHz_us(uint32_t val){

// input checking
    // max input is 349,524
    if (val > 349524){
        printf("\nDelay_48MHz_us delay out of bounds %i\n", val);
        exit(1);
    }

    // Local Variables
// These values are used to timeout the transitions
uint32_t delay_cnt;

// Calculate the required number of clock cycles
// 48MHz --> 20.8333ns/clk
// # of clocks per us = 1e-6 * 48e6 = 48
// required number of clocks = val * clks/us
delay_cnt = val * 48;

// Setup the systick timer
SysTick->CTRL = 0x0005;      // set clk to CPU clk (bit 2) and enable (bit 0)
SysTick->LOAD = delay_cnt;
SysTick->VAL = 0x0;

// Wait
    while(!(SysTick->CTRL & 0x00010000))
    ;

// Done waiting
return 0;
} // end Delay_48MHz_us
```

Figure 2-48. STRVR Register

31-24	23-0
RESERVED	RELOAD
R/W	R/W

Table 2-55. STRVR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	RESERVED	R/W	Undefined	
23-0	RELOAD	R/W	Undefined	Value to load into the SysTick Current Value Register when the counter reaches 0.

Figure 2-49. STCVR Register

31-24	23-0
RESERVED	CURRENT
R/W	R/W

Table 2-56. STCVR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	RESERVED	R/W	Undefined	
23-0	CURRENT	R/W	Undefined	Current value at the time the register is accessed. No read-modify-write protection is provided, so change with care. Writing to it with any value clears the register to 0. Clearing this register also clears the COUNTFLAG bit of the SysTick Control and Status Register.

Figure 2-47. STCSR Register

31	30	29	28	27	26	25	24
RESERVED							
R/W-0h							
23	22	21	20	19	18	17	16
RESERVED							COUNTFLAG
R/W-0h							R-0h
15	14	13	12	11	10	9	8
RESERVED							
R/W-0h							
7	6	5	4	3	2	1	0
RESERVED					CLKSOURCE	TICKINT	ENABLE
R/W-0h					R-1h	R/W-0h	R/W-0h

Table 2-54. STCSR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-17	RESERVED	R/W	0h	
16	COUNTFLAG	R	0h	Returns 1 if timer counted to 0 since last time this was read. Clears on read by application of any part of the SysTick Control and Status Register. If read by the debugger using the DAP, this bit is cleared on read-only if the MasterType bit in the AHB-AP Control Register is set to 0. Otherwise, the COUNTFLAG bit is not changed by the debugger read.
15-3	RESERVED	R/W	0h	
2	CLKSOURCE	R	1h	Clock source. 0b = Not applicable 1b = Core clock
1	TICKINT	R/W	0h	
0	ENABLE	R/W	0h	Enable SysTick counter 0b (R/W) = Counter disabled

ARM M4 Peripherals

- System Control Block – SCB
  - Access to a range of system level registers

ARM M4 Peripherals

- SCB - Registers

Table 2-58. SCB Registers

Offset	Acronym	Register Name	Type	Reset	Section
D00h	CPUID	CPUID Base Register	read-only	410FC241h	Section 2.4.5.1
D04h	ICSR	Interrupt Control State Register	read-write	00000000h	Section 2.4.5.2
D08h	VTOR	Vector Table Offset Register	read-write	00000000h	Section 2.4.5.3
D0Ch	AIRCR	Application Interrupt/Reset Control Register	read-write	FA050000h	Section 2.4.5.4
D10h	SCR	System Control Register	read-write	00000000h	Section 2.4.5.5
D14h	CCR	Configuration Control Register	read-write	00000200h	Section 2.4.5.6
D18h	SHPR1	System Handlers 4-7 Priority Register	read-write	00000000h	Section 2.4.5.7
D1Ch	SHPR2	System Handlers 8-11 Priority Register	read-write	00000000h	Section 2.4.5.8
D20h	SHPR3	System Handlers 12-15 Priority Register	read-write	00000000h	Section 2.4.5.9
D24h	SHCSR	System Handler Control and State Register	read-write	00000000h	Section 2.4.5.10
D28h	CFSR	Configurable Fault Status Registers	read-write	00000000h	Section 2.4.5.11
D2Ch	HFSR	Hard Fault Status Register	read-write	00000000h	Section 2.4.5.12
D30h	DFSR	Debug Fault Status Register	read-write	00000000h	Section 2.4.5.13
D34h	MMFAR	Mem Manage Fault Address Register	read-write	Undefined	Section 2.4.5.14
D38h	BFAR	Bus Fault Address Register	read-write	Undefined	Section 2.4.5.15
D3Ch	AFSR	Auxiliary Fault Status Register	read-write	00000000h	Section 2.4.5.16
D40h	PFR0	Processor Feature register0	read-only	00000030h	Section 2.4.5.17
D44h	PFR1	Processor Feature register1	read-only	00000200h	Section 2.4.5.18
D48h	DFR0	Debug Feature register0	read-only	00100000h	Section 2.4.5.19
D4Ch	AFR0	Auxiliary Feature register0	read-only	00000000h	Section 2.4.5.20
D50h	MMFR0	Memory Model Feature register0	read-only	00100030h	Section 2.4.5.21
D54h	MMFR1	Memory Model Feature register1	read-only	00000000h	Section 2.4.5.22
D58h	MMFR2	Memory Model Feature register2	read-only	01000000h	Section 2.4.5.23
D5Ch	MMFR3	Memory Model Feature register3	read-only	00000000h	Section 2.4.5.24
D60h	ISAR0	ISA Feature register0	read-only	01101110h	Section 2.4.5.25
D64h	ISAR1	ISA Feature register1	read-only	02112000h	Section 2.4.5.26
D68h	ISAR2	ISA Feature register2	read-only	21232231h	Section 2.4.5.27
D6Ch	ISAR3	ISA Feature register3	read-only	01111131h	Section 2.4.5.28
D70h	ISAR4	ISA Feature register4	read-only	01310132h	Section 2.4.5.29
D88h	CPACR	Coprocessor Access Control Register	read-write	00F00000h	Section 2.4.5.30

SCB->CPUID

…

SCB->SHP[1] – SHP[3]

SCB->PFR[0 – PFR[1]

SCB->MMFR[0 – MMFR[3]

SCB>ISAR[0] - ISAR[4]

ARM M4 Peripherals

- SCB – CPU ID Register

SCB->CPUID

Figure 2-51. CPUID Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
IMPLEMENTER								VARIANT				CONSTANT			
R-41h								R-0h				R-Fh			
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PARTNO												REVISION			
R-C24h												R-1h			

Table 2-59. CPUID Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	IMPLEMENTER	R	41h	Implementor code.
23-20	VARIANT	R	0h	Implementation defined variant number.
19-16	CONSTANT	R	Fh	Reads as 0xC
15-4	PARTNO	R	C24h	Number of processor within family.
3-0	REVISION	R	1h	Implementation defined revision number.

ARM M4 Peripherals

- SCB – Interrupt Control State Register

SCB->ICSR

Figure 2-52. ICSR Register

31	30	29	28	27	26	25	24
NMIPENDSET	RESERVED		PENDSVSET	PENDSVCLR	PENDSTSET	PENDSTCLR	RESERVED
R/W-0h	R/W-0h		R/W-0h	W-0h	R/W-0h	W-0h	R/W-0h
23	22	21	20	19	18	17	16
ISRPREEMPT	ISRPENDING	RESERVED				VECTPENDING	
R-0h	R-0h	R/W-0h				R-0h	
15	14	13	12	11	10	9	8
VECTPENDING				RETTOBASE	RESERVED		VECTACTIVE
R-0h				R-0h	R/W-0h		R-0h
7	6	5	4	3	2	1	0
VECTACTIVE							
R-0h							

• • •

Bit	Field	Type	Reset	Description
17-12	VECTPENDING	R	0h	Pending ISR number field. VECTPENDING contains the interrupt number of the highest priority pending ISR.
11	RETTOBASE	R	0h	This bit is 1 when the set of all active exceptions minus the IPSR_current_exception yields the empty set.
10-9	RESERVED	R/W	0h	
8-0	VECTACTIVE	R	0h	Active ISR number field. Reset clears the VECTACTIVE field.

ARM M4 Peripherals

- Memory Protection Unit – MPU
  - The MPU breaks the processor memory into 8 logical regions
  - Each region has settings to:
    - Set start address and size
    - Limit access (privilege)
    - Set as bufferable
    - Set as cacheable
    - Set as non-executable

ARM M4 Peripherals

- MPU - Privilege
  - Program memory, data memory, and peripheral access limitations
  - Operating System (privileged)
    - Access to all of program memory and data memory
    - Access to all peripherals
  - Application Program A (unprivileged)
    - Access to portions of program memory and data memory
    - Access to some peripherals
  - Application Program B (unprivileged)
    - Access to portions of program memory and data memory
    - Access to some peripherals

ARM M4 Peripherals

- MPU - Registers

MPU->TYPE

...

Offset	Acronym	Register Name	Type	Reset
D90h	TYPE	MPU Type Register	read-only	00000800h
D94h	CTRL	MPU Control Register	read-write	00000000h
D98h	RNR	MPU Region Number Register	read-write	00000000h
D9Ch	RBAR	MPU Region Base Address Register	read-write	00000000h
DA0h	RASR	MPU Region Attribute and Size Register	read-write	00000000h
DA4h	RBAR_A1	MPU Alias 1 Region Base Address register	read-write	00000000h
DA8h	RASR_A1	MPU Alias 1 Region Attribute and Size register	read-write	00000000h
DACh	RBAR_A2	MPU Alias 2 Region Base Address register	read-write	00000000h
DB0h	RASR_A2	MPU Alias 2 Region Attribute and Size register	read-write	00000000h
DB4h	RBAR_A3	MPU Alias 3 Region Base Address register	read-write	00000000h
DB8h	RASR_A3	MPU Alias 3 Region Attribute and Size register	read-write	00000000h

ARM M4 Peripherals

- Floating Point Unit – FPU
  - 32-bit instructions for single-precision (C float) data-processing operations
  - Combined multiply and accumulate instructions for increased precision (Fused MAC)
  - Hardware support for conversion, addition, subtraction, multiplication with optional accumulate, division, and square root
  - Hardware support for denormals and all IEEE rounding modes
  - 32 dedicated 32-bit single-precision registers, also addressable as 16 double-word registers
  - Decoupled three stage pipeline

ARM M4 Peripherals

- Floating Point Unit – FPU
  - Extended Register Set
    - Sixteen 64-bit double-word registers, D0-D15
    - Thirty-two 32-bit single-word registers, S0-S31
    - A combination of registers from the above views
  - The FPU is disabled from reset. You must enable it before you can use any floating-point instructions.
    - In many compilers, such as in TI's Code Composer Studio IDE, if the hardware FPU option is selected in the compiler setting, the initialization code enables the FPU before entering the main() function. In this case, users do not need to manually turn on the FPU in the main application code.
    - The processor must be in privileged mode to read from and write to the Coprocessor Access Control (CPAC) register – see the spec for details

ARM M4 Peripherals

- Floating Point Unit – Registers

Offset	Acronym	Register Name	Type	Reset
F34h	FPCCR	Floating Point Context Control Register	read-write	C0000000h
F38h	FPCAR	Floating-Point Context Address Register	read-write	00000000h
F3Ch	FPDSCR	Floating Point Default Status Control Register	read-write	00000000h
F40h	MVFR0	Media and FP Feature Register 0 (MVFR0)	read-only	10110021h
F44h	MVFR1	Media and FP Feature Register 1 (MVFR1)	read-only	11000011h

FPU->FPCCR

...

FPU->MVFR1

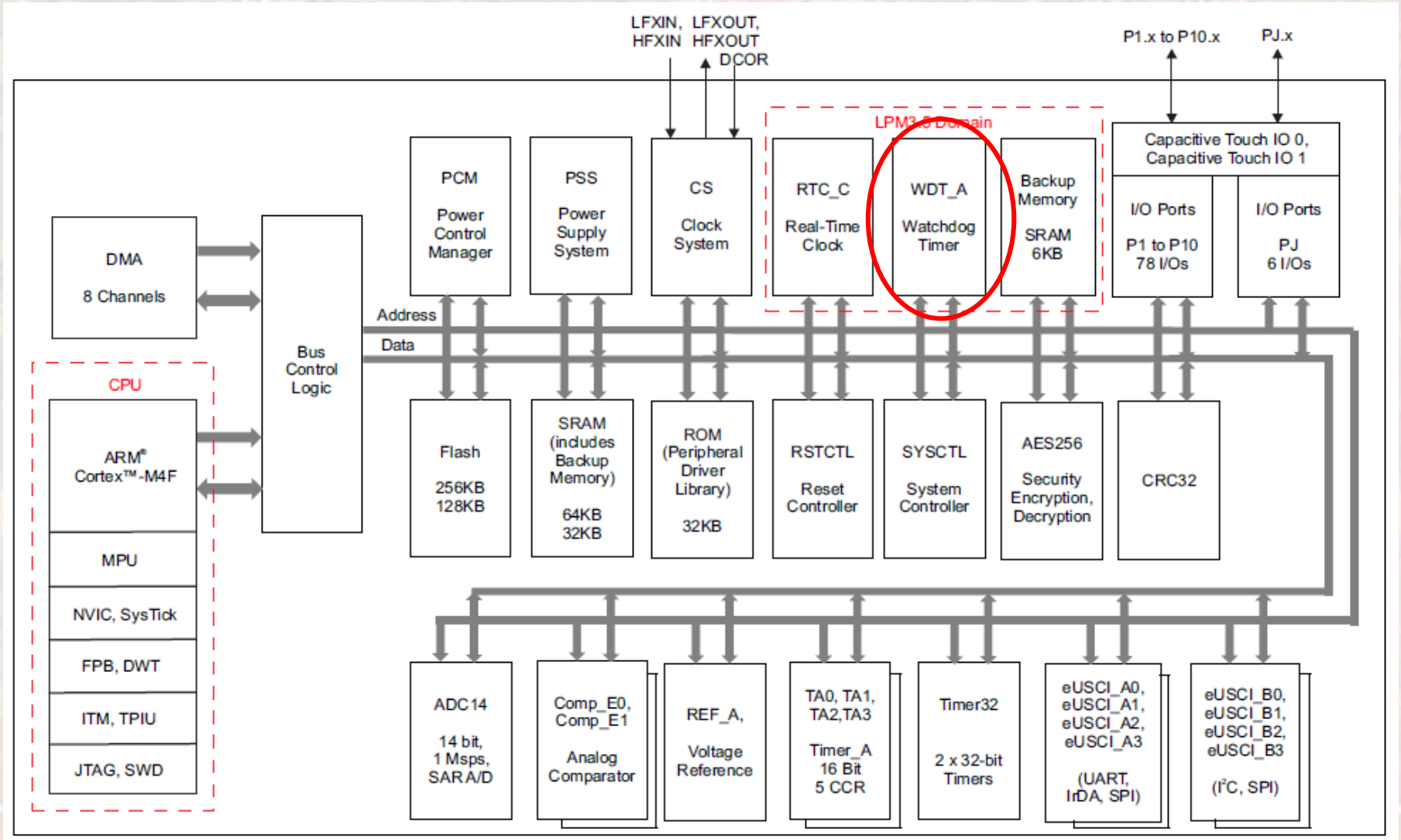
ARM M4 Peripherals

- Debug Peripherals
  - FPB - Flash Patch and Breakpoint
    - Causes flashed code to be re-directed to a different location
    - Causes flashed code to cause a breakpoint in hardware
  - DWT – Data Watchpoint and Trace
    - Hardware triggers
    - Data Address triggers
    - PC Sampler Triggers
  - ITM – Instrumentation Trace Macrocell

    TPIU – Trace Port Interface Unit
    - Provide printf style debugging to an output stream
  - JTAG – Joint Test Action Group

    SWD – Serial Wire Debug
    - Access to the debug peripherals

ARM M4 Peripherals



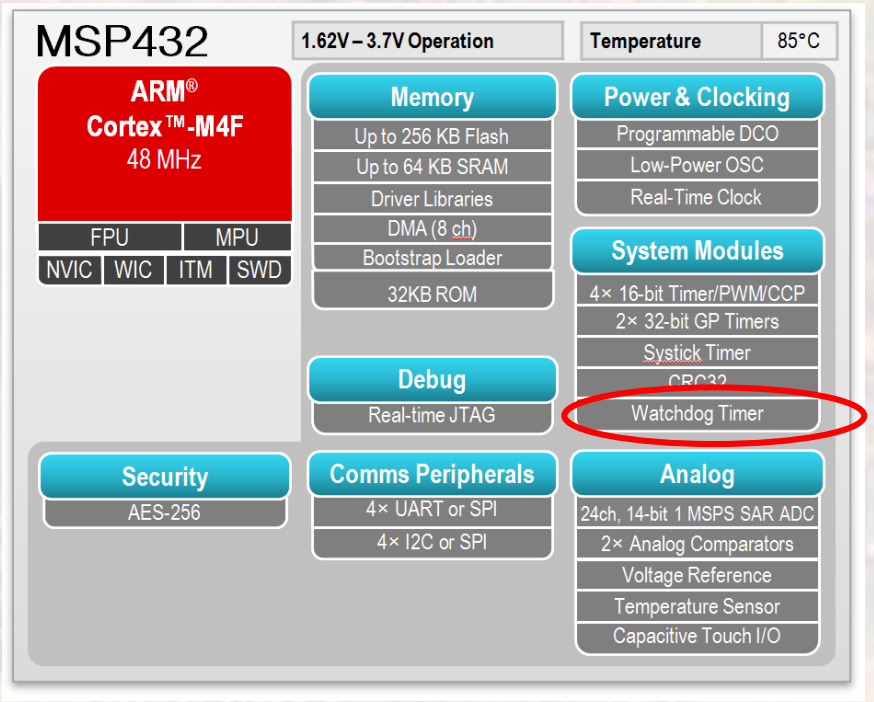
Watchdog Timer

- Basic Watchdog Function
  - What happens if your code gets trapped in a loop or runs off into the weeds?
  - Checks to make sure your software is running
  - Count up timer that creates a system reset if it gets to a programmed value
  - System software must periodically reset the timer to prevent it from activating a system reset

Watchdog Timer

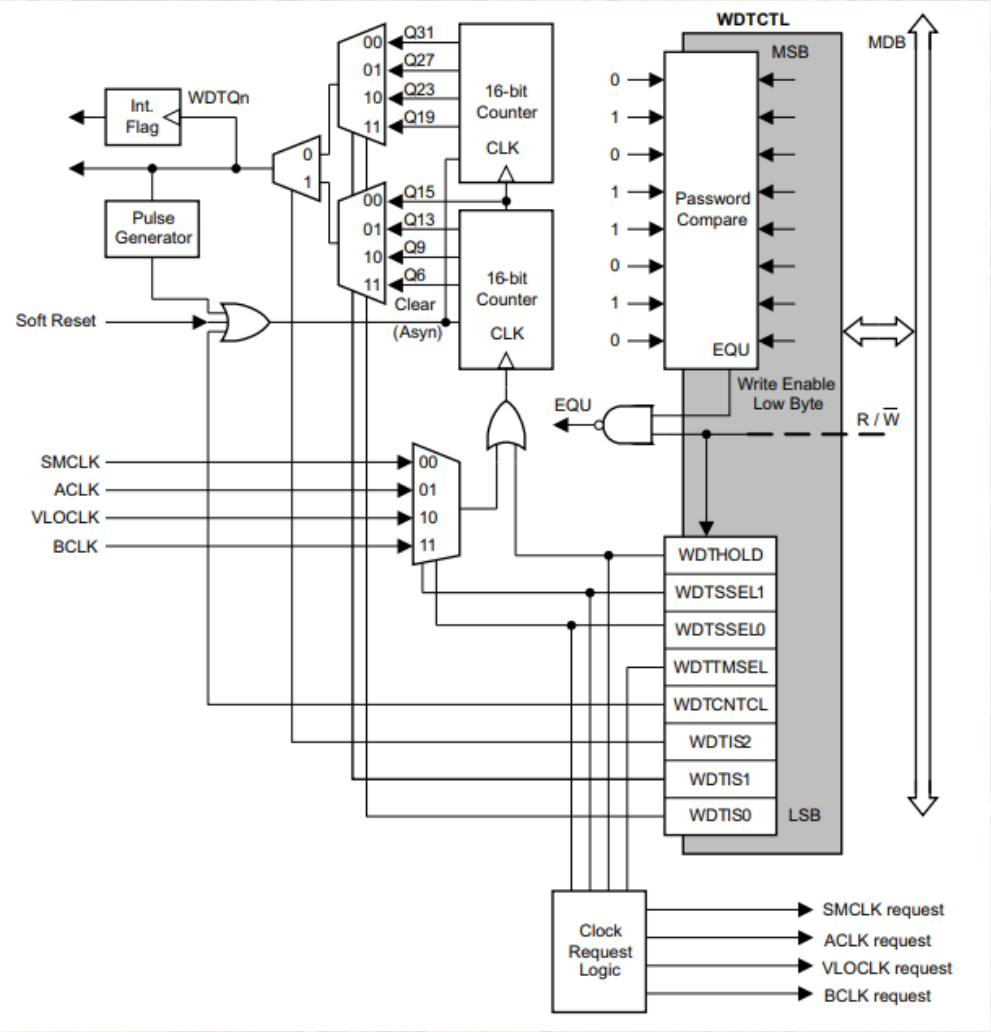
- MSP432 Watchdog Timer
  - ARM (AMBA Compliant) timer
  - 32 bit timer
  - Password protected
    - Any violation results in a system reset
  - Can be configured as an interval counter

  - COUNT UP TIMER



Watchdog Timer

- MSP432 Watchdog Timer



Watchdog Timer

- MSP432 Watchdog Timer
  - Password Protection
    - 16 bit control register
    - Uses special instruction to access
    - Upper byte must be 0x5A (password) for writes
    - Upper byte returns 0x69 for reads
    - Any incorrect instruction access → system reset
    - Any incorrect password → system reset

Watchdog Timer

- MSP432 Watchdog Timer
  - Reset Defaults
    - WD Timer not active
    - Count set to 2^{15}
    - Clock source set to SMCLK

Watchdog Timer

- MSP432 Watchdog Timer
  - Interval Timer Mode
    - Count up counter
    - Sets a flag or creates an interrupt when the programmed count is reached

Watchdog Timer

- MSP432 Watchdog Timer
  - Watchdog Control Register WDT_A->CTL

Figure 15-2. WDTCTL Register

15	14	13	12	11	10	9	8
WDTPW							
rw-{0}	rw-{1}	rw-{1}	rw-{0}	rw-{1}	rw-{0}	rw-{0}	rw-{1}
7	6	5	4	3	2	1	0
WDTHOLD	WDTSSEL		WDTTMSEL	WDTCNTCL	WDTIS		
rw-{0}	rw-{0}	rw-{0}	rw-{0}	rw-{0}	rw-{1}	rw-{0}	rw-{0}

Watchdog Timer

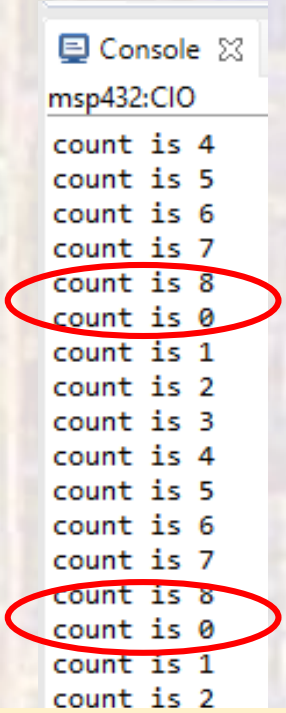
- MSP432 Watchdog Timer
  - Watchdog Control Register

Table 15-3. WDTCTL Register Description[1]

Bit	Field	Type	Reset	Description
15-8	WDTPW	RW	69h	Watchdog timer password. Always read as 069h. Must be written as 05Ah, or the WDT generates a reset.
7	WDTHOLD	RW	0h	Watchdog timer hold. This bit stops the watchdog timer. Setting WDTHOLD = 1 when the WDT is not in use conserves power. 0b = Watchdog timer is not stopped 1b = Watchdog timer is stopped
6-5	WDTSSEL	RW	0h	Watchdog timer clock source select 00b = SMCLK 01b = ACLK 10b = VLOCLK 11b = BCLK
4	WDTTMSEL	RW	0h	Watchdog timer mode select 0b = Watchdog mode[2] 1b = Interval timer mode[3]
3	WDTCNTCL	W	0h	Watchdog timer counter clear. Setting WDTCNTCL = 1 clears the count value to 0000h. This bit always reads 0h 0b = No action 1b = WDTCNT = 0000h
2-0	WDTIS	RW	4h	Watchdog timer interval select. These bits select the watchdog timer interval to generate either a WDT interrupt or a WDT reset . 000b = Watchdog clock source / 2^{31} (18:12:16 at 32.768 kHz) 001b = Watchdog clock source / 2^{27} (01:08:16 at 32.768 kHz) 010b = Watchdog clock source / 2^{23} (00:04:16 at 32.768 kHz) 011b = Watchdog clock source / 2^{19} (00:00:16 at 32.768 kHz) 100b = Watchdog clock source / 2^{15} (1 s at 32.768 kHz) 101b = Watchdog clock source / 2^{13} (250 ms at 32.768 kHz) 110b = Watchdog clock source / 2^{9} (15.625 ms at 32.768 kHz) 111b = Watchdog clock source / 2^{5} (1.95 ms at 32.768 kHz)

Watchdog Timer

- MSP432 W
  - Example



WD Timer resets the processor

```
/*
 * wdt_example.c
 *
 *  Created on: Jul 24, 2018
 *      Author: johnsontimoj
 */
//////////////////////////
//
// example file for showing watch dog timer operation
//
// kills a counting program after a period of time
//
// inputs: none
// outputs: count and failure
//
//////////////////////////

// includes
#include "msp.h"
#include <stdio.h>

int main(void){

   // setup watchdog
   // 0101 1010     0         0        0        0       010
   // password   hold off   sm clk  wd mode   no clr   2**23 cnts(~8M clocks)
   WDT_A->CTL = 0x5A02;

   int i;
   for(i = 0; i<20; i++){
       printf("count is %i\n", i);
       __delay_cycles(1000000);
   }

   return 0;
}
```

Watchdog Timer

- MSP432 Watchdog Timer
  - Example

```
msp432:CIO
[CORTEX_M4_0] count is 0
count is 1
count is 2
count is 3
count is 4
count is 5
count is 6
count is 7
count is 8
count is 9
count is 10
count is 11
count is 12
count is 13
count is 14
count is 15
```

```c
// includes
#include "msp.h"
#include <stdio.h>

int main(void){

   // setup watchdog
   // 0101 1010     0         0        0        0       010
   // password   hold off   sm clk  wd mode   no clr   2**23 cnts(~8M clocks)
   WDT_A->CTL = 0x5A02;

   int i;
   for(i = 0; i<20; i++){
       // clear timer
       // returns 0x69xx, need to write 0x5Ax(1bbb)
       // turn off bits 1 in 6, 0 in 9
       // turn on bits 0 in 5, bit 1 in A
       // turn on bit 3 in lsb nibble
       WDT_A->CTL = (WDT_A->CTL & ~0x2100) | 0x1208;

       printf("count is %i\n", i);
       __delay_cycles(1000000);
   }

   return 0;
}
```

Resets the WD timer