Laboratorio 10

R. Ferrero, P. Bernardi

Politecnico di Torino

Dipartimento di Automatica e Informatica (DAUIN)

Torino - Italy

Diagramma a blocchi (pag. 9)

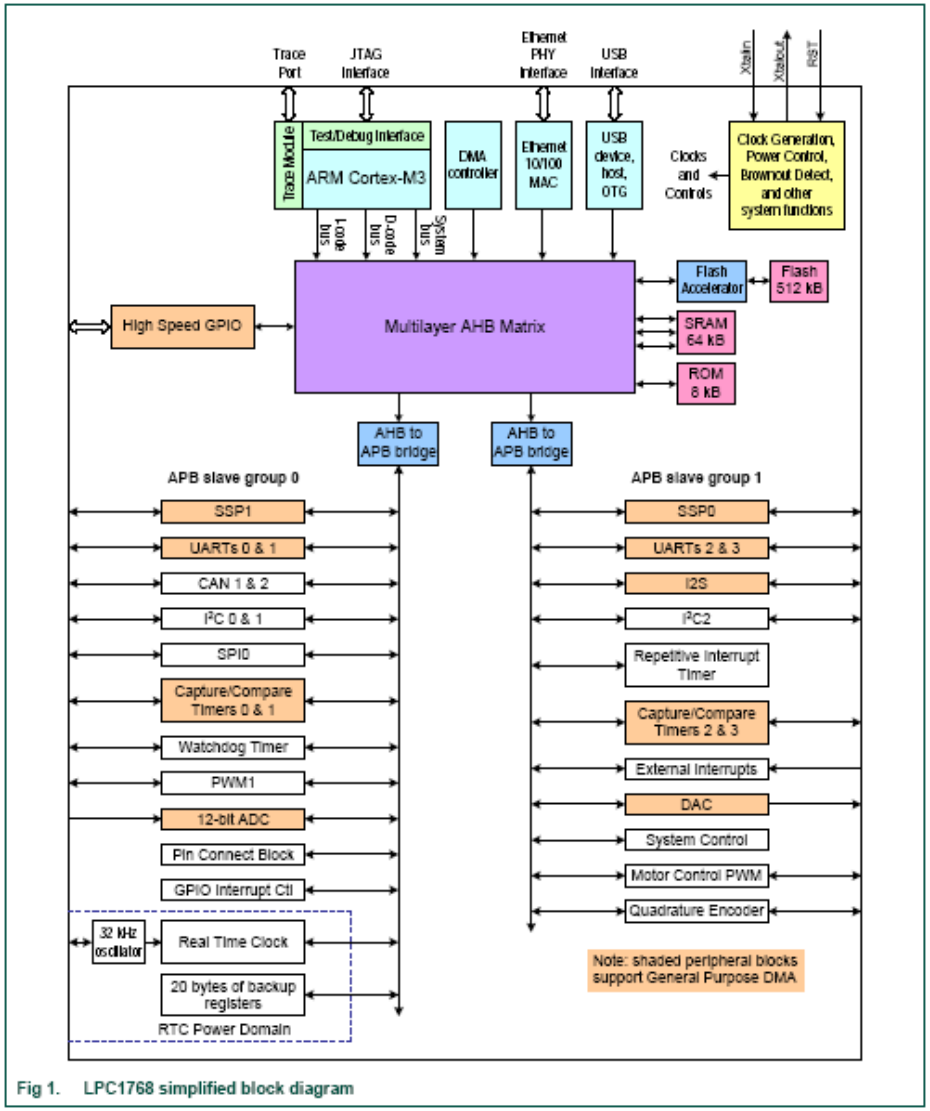
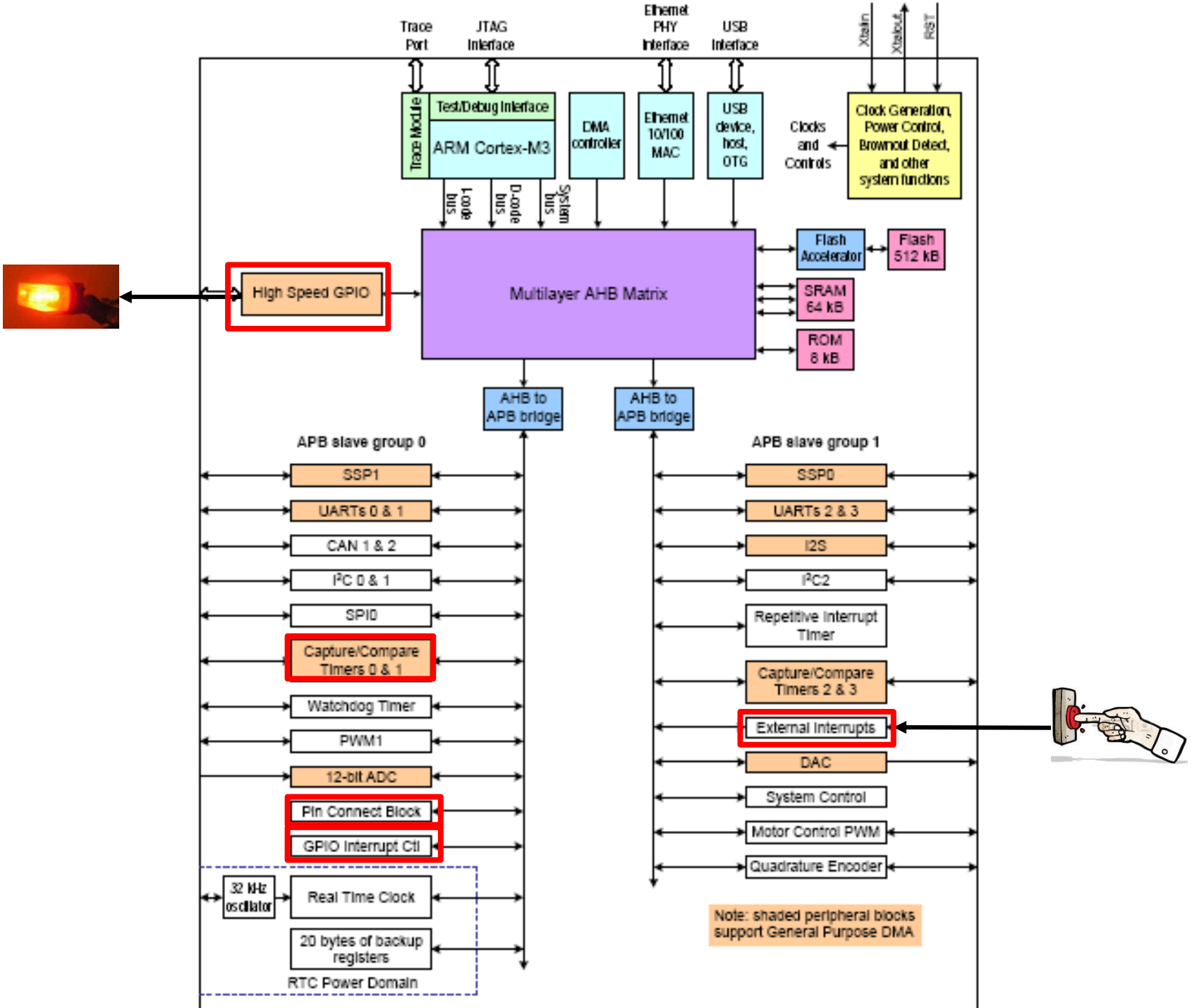


Fig 1. LPC1768 simplified block diagram


Trace Port
JTAG Interface
Ethernet PHY Interface
USB Interface
Xtalin
Xtalout
RST
Trace Module
Test/Debug Interface
ARM Cortex-M3
DMA controller
Ethernet 10/100 MAC
USB device, host, OTG
Clocks and Controls
Clock Generation, Power Control, Brownout Detect, and other system functions
I-code bus
D-code bus
System bus
High Speed GPIO
Multilayer AHB Matrix
Flash Accelerator
Flash 512 kB
SRAM 64 kB
ROM 8 kB
AHB to APB bridge
AHB to APB bridge
APB slave group 0
SSP1
UARTs 0 & 1
CAN 1 & 2
I²C 0 & 1
SPI0
Capture/Compare Timers 0 & 1
Watchdog Timer
PWM1
12-bit ADC
Pin Connect Block
GPIO Interrupt Ctl
32 kHz oscillator
Real Time Clock
20 bytes of backup registers
RTC Power Domain
APB slave group 1
SSP0
UARTs 2 & 3
I2S
I²C2
Repetitive Interrupt Timer
Capture/Compare Timers 2 & 3
External Interrupts
DAC
System Control
Motor Control PWM
Quadrature Encoder
Note: shaded peripheral blocks support General Purpose DMA

Main registers

Table 426. TIMER/COUNTER0-3 register map

Generic Name	Description	Access	Reset Value[1]	TIMERn Register/ Name & Address
IR	Interrupt Register. The IR can be written to clear interrupts. The IR can be read to identify which of eight possible interrupt sources are pending.	R/W	0	T0IR - 0x4000 4000 T1IR - 0x4000 8000 T2IR - 0x4009 0000 T3IR - 0x4009 4000
TCR	Timer Control Register. The TCR is used to control the Timer Counter functions. The Timer Counter can be disabled or reset through the TCR.	R/W	0	T0TCR - 0x4000 4004 T1TCR - 0x4000 8004 T2TCR - 0x4009 0004 T3TCR - 0x4009 4004
TC	Timer Counter. The 32-bit TC is incremented every PR+1 cycles of PCLK. The TC is controlled through the TCR.	R/W	0	T0TC - 0x4000 4008 T1TC - 0x4000 8008 T2TC - 0x4009 0008 T3TC - 0x4009 4008
MCR	Match Control Register. The MCR is used to control if an interrupt is generated and if the TC is reset when a Match occurs.	R/W	0	T0MCR - 0x4000 4014 T1MCR - 0x4000 8014 T2MCR - 0x4009 0014 T3MCR - 0x4009 4014
MR0	Match Register 0. MR0 can be enabled through the MCR to reset the TC, stop both the TC and PC, and/or generate an interrupt every time MR0 matches the TC.	R/W	0	T0MR0 - 0x4000 4018 T1MR0 - 0x4000 8018 T2MR0 - 0x4009 0018

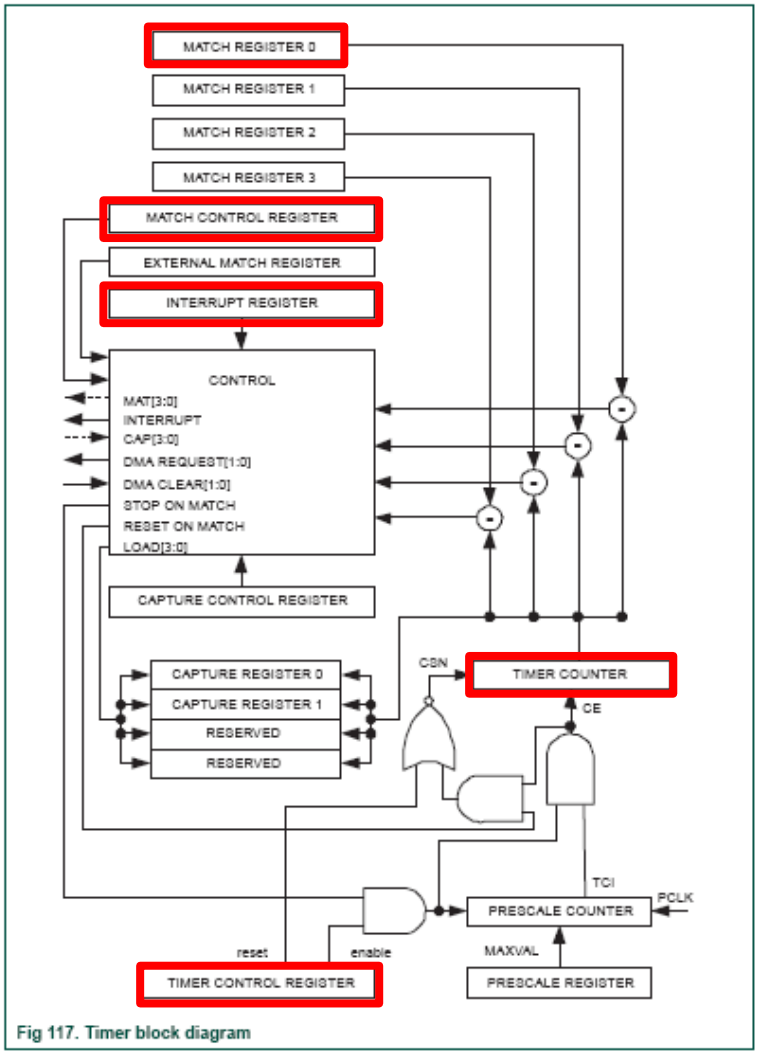


Fig 117. Timer block diagram

Timer counter

- The Timer Counter counts clock cycles.
- Unless it is reset before reaching its upper limit, the Timer Counter will count up through the value 0xFFFF FFFF and then wrap back to the value 0x0000 0000.

Timer Control Register

- The lowest two bits of TCR control the operation of the Timer/Counter.
- Bit 0 = 1 -> enable timer for counting.
- Bit 1 = 1 -> reset counter. The value of the timer is fixed until the value of this bit is 1.

Table 428. Timer Control Register (TCR, TIMERn: TnTCR - addresses 0x4000 4004, 0x4000 8004, 0x4009 0004, 0x4009 4004) bit description

Bit	Symbol	Description	Reset Value
0	Counter Enable	When one, the Timer Counter and Prescale Counter are enabled for counting. When zero, the counters are disabled.	0
1	Counter Reset	When one, the Timer Counter and the Prescale Counter are synchronously reset on the next positive edge of PCLK. The counters remain reset until TCR[1] is returned to zero.	0
31:2	-	Reserved, user software should not write ones to reserved bits. The value read from a reserved bit is not defined.	NA

Match Registers

- The Match register values are continuously compared to the Timer Counter value.
- The following formula can be used to compute the number of clock cycles to count

  count = time [s] * frequency [1/s]

- In the sample project, the function `SystemInit()` sets the frequency of the timer to 25MHz.

Match Control Register

- The Match Control Register is used to control what operations are performed when one of the Match Registers matches the Timer Counter

Table 430. Match Control Register (T[0/1/2/3]MCR - addresses 0x4000 4014, 0x4000 8014, 0x4009 0014, 0x4009 4014) bit description

Bit	Symbol	Value	Description	Reset Value
0	MR0I	1	Interrupt on MR0: an interrupt is generated when MR0 matches the value in the TC.	0
		0	This interrupt is disabled	
1	MR0R	1	Reset on MR0: the TC will be reset if MR0 matches it.	0
		0	Feature disabled.	
2	MR0S	1	Stop on MR0: the TC and PC will be stopped and TCR[0] will be set to 0 if MR0 matches the TC.	0
		0	Feature disabled.	

Interrupt Register

- The Interrupt Register consists of
  - 4 bits for the match interrupts and
  - 4 bits for the capture interrupts.
- If an interrupt is generated, the corresponding bit in the IR is high, otherwise the bit is low.
- Writing 1 to the corresponding IR bit will reset the interrupt, writing 0 has no effect.

Table 427. Interrupt Register (T[0/1/2/3]IR - addresses 0x4000 4000, 0x4000 8000, 0x4009 0000, 0x4009 4000) bit description

Bit	Symbol	Description	Reset Value
0	MR0 Interrupt	Interrupt flag for match channel 0.	0
1	MR1 Interrupt	Interrupt flag for match channel 1.	0
2	MR2 Interrupt	Interrupt flag for match channel 2.	0
3	MR3 Interrupt	Interrupt flag for match channel 3.	0
4	CR0 Interrupt	Interrupt flag for capture channel 0 event.	0
5	CR1 Interrupt	Interrupt flag for capture channel 1 event.	0
31:6	-	Reserved	-

Esercizio: Simon

- Implementare il gioco Simon.
- Usando i pulsanti Key1, Key2, Int0 il giocatore deve ripetere una sequenza casuale, mostrata tramite i led 4, 5, 6.
- Se la sequenza di pulsanti premuti dal giocatore è uguale alla sequenza con cui i led si sono illuminati, il giocatore ha vinto e il gioco si ripete incrementando di 1 la sequenza.
- Altrimenti il giocatore perde e la sequenza riparte da uno.

Sequenza mostrata con i led

```
// n è la lunghezza della sequenza
i = 0;
while (i < n) {
    v = numero casuale fra 0 e 2;
    //0: led 4, 1: led 5, 2: led 6
    accendere il led per 1,5 s;
    spegnere il led per 1,5 s;
    i ++;
}
```

Generazione di numeri casuali

- Per ottenere numeri casuali si può controllare il valore del timer counter.
- Calcolando il modulo 3 del timer counter, si ottiene un numero casuale fra 0 e 2.
- Si consiglia di utilizzare un timer diverso da quello utilizzato per gestire l’accensione e lo spegnimento dei led.
- Il timer counter può essere controllato anche per verificare un rimbalzo del pulsante.

Esito della partita

- Se il giocatore ha premuto i pulsanti nella sequenza corretta, si accendono opportunamente i led 4-11 per mostrare il valore di `n` in binario (il led 11 corrisponde al led meno significativo). Il gioco riparte con `n` = `n` + 1 quando l'utente preme un pulsante.
- Se invece l'`i`-esimo pulsante premuto dall'utente è sbagliato, sui led 4-11 è mostrato il valore `i`. Il gioco riparte con `n` = 1 quando il giocatore preme un pulsante.

Suggerimenti

- Per mostrare il numero binario `x` sui led 4-11, è sufficiente assegnare agli 8 bit bassi di LPC_GPIO2->FIOPIN il valore `x`.
- L'handler del timer deve gestire l'accensione dei led con una sequenza casuale e memorizzare la sequenza.
- L'handler dei pulsanti deve verificare se il pulsante premuto corrisponde all'elemento i-esimo della sequenza memorizzata.

Suggerimenti (cont.)

- Si consiglia di usare una variabile globale, comune agli handler di timer e pulsanti, per tenere traccia dello stato corrente durante la partita.
- Il diagramma di stato è mostrato nella successiva slide.

Diagramma di stato