**Chapter 8: Timers**

**8.1 Introduction . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 236**

8.1.1 System Diagram . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .237

8.1.2 Notices . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .237

**8.2 CPU Private Timers and Watchdog Timers . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 238**

8.2.1 Clocking . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .238

8.2.2 Interrupt to PS Interrupt Controller . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .238

8.2.3 Resets. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .238

8.2.4 Register Overview . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .238

**8.3 Global Timer (GT) . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 239**

8.3.1 Clocking . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .239

8.3.2 Register Overview . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .239

**8.4 System Watchdog Timer (SWDT). . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 240**

8.4.1 Features . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .240

8.4.2 Block Diagram . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .241

8.4.3 Functional Description . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .241

8.4.4 Register Overview . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .242

8.4.5 Programming Model. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .243

8.4.6 Clock Input Option for SWDT . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .243

8.4.7 Reset Output Option for SWDT . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .243

**8.5 Triple Timer Counters (TTC) . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 244**

8.5.1 Features . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .244

8.5.2 Block Diagram . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .244

8.5.3 Functional Description . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .245

8.5.4 Register Overview . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .246

8.5.5 Programming Model. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .247

8.5.6 Clock Input Option for Counter/Timer . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .248

**8.6 I/O Signals . . . . . .. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 249**

**8.1 Introduction**

**소개**

Each Cortex-A9 processor has its own private 32-bit timer and 32-bit watchdog timer.

Cortex-A9 각각 전용의 개인적인 32비트 타이머와 32비트 워치독 타이머를 가진다.

Both processors share a global 64-bit timer. These timers are always clocked at 1/2 of the CPU frequency (CPU\_3x2x).

두 프로세서 모두 전역64비트 타이머를 공유한다. 이 타이머들은 항상 CPU 주파수(CPU\_3x2x)의 2분의 1 클록 된다.

On the system level, there is a 24-bit watchdog timer and two 16-bit triple timer/counters.

시스템레벨에서는, 24비트 워치독 타이머와 2개의 16비트 트리플 카운터가 있다.

The system watchdog timer is clocked at 1/4 or 1/6 of the CPU frequency (CPU\_1x), or can be clocked by an external signal from an MIO pin or from the PL.

시스템 워치독 타이머는 CPU주파수(CPU\_1x)의 6분의1 또는 4분의 1 클록 된다.

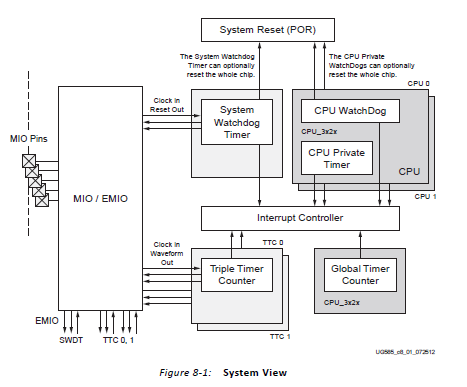
The two triple timers/counters are always clocked at 1/4 or 1/6 of the CPU frequency (CPU\_1x), and are used to count the widths of signal pulses from an MIO pin or from the PL.

두개의 트리플 타이머/카운터는 항상 CPU주파수(CPU\_1x)의 6분의1 또는 4분의 1 클록되고 PL 또는 MIO핀에서 신호펄스의 가로를 카운트 하기 위해 사용된다.

**8.1.1 System Diagram**

The relationships of the system timers are shown in Figure 8-1.

시스템 타이머 관계



**8.1.2 Notices**

**7z007s and 7z007s CLG225 Devices**

The 7z007s single core and 7z010 dual core CLG225 devices support 32 MIO pins (not 54).

7z007s single core 와 7z010 dual core CLG225 장치들은 32MIO 핀을 지원한다. (54개아님)

This is shown in the MIO table in section 2.5.4 MIO-at-a-Glance Table.

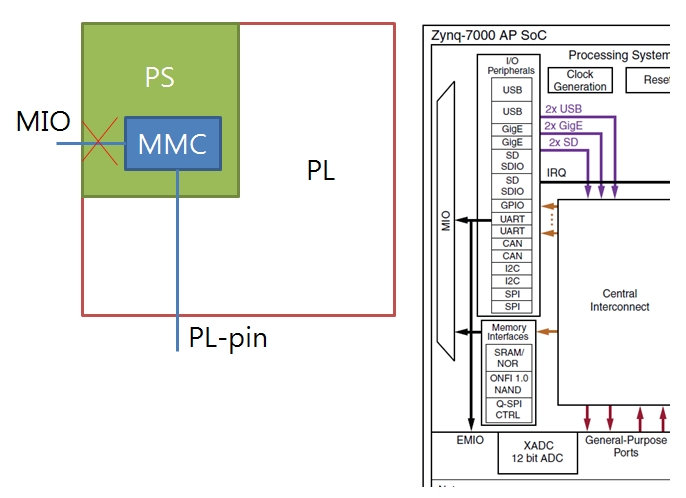
MIO 표를 2.5.4 MIO-at-a-Glance Table가 보여준다.

The 7z007s and 7z010 CLG225 devices restrict the available MIO pins so connections through the EMIO might need to be considered.

The 7z007s 와 7z010 CLG225 장치들은 사용 가능한 MIO핀을 제한하므로 EMIO를 통한 연결을 고려해야 할 수 있습니다.

All of the 7z007s and 7z010 CLG225 device restrictions are listed in section

1.1.3 Notices.



zynq칩은 우리가 알고 있는 GPIO 라고 하는 핀을 MIO 라고 명명한다.

이 MIO는 GPIO 나 SPI 버스, MII 버스, NAND 등의 컨트롤러핀으로 사용한다.

관련한 레지스터를 조작하여 해당 컨트롤러 핀으로 사용한다.

MIO는 내부 하드웨어 블럭 전용의 핀인이며 각 MIO핀들은 칩의 특정 핀으로 연결된다.

MIO 핀들은 여러 하드웨어 블럭이 사용하다 보니 여러 하드웨어블럭을 사용하고자 할때 MIO 핀이 모자라게 된다.

이럴 때 PS 내부의 하드웨어블럭(MAC, SPI, MMC등)을 PL 블럭의 핀으로 뽑아 사용할 수 있다. 이럴 때 EMIO 라 한다.

역시 Program Logic 의 힘이다.

**8.2 CPU Private Timers and Watchdog Timers**

The CPU private timers and watchdog timers are fully documented in the *Cortex-A9 MPCore*

*Technical Requirements Document*, sections 4.1 and 4.2 (see Appendix A, Additional Resources).  
CPU 개인 타이머 및 워치 독 타이머는 Cortex-A9 MPCore에 완벽하게 설명되어 있습니다. 기술 요구 사항 문서, 섹션 4.1 및 4.2 (부록 A, 추가 리소스 참조).

Both the timer and watchdog blocks have the following features:

타이머 및 워치독 블록은 다음과 같은 기능을 가진다.

• 32-bit counter that generates an interrupt when it reaches zero

0에 도달할 때 인터럽트를 생성하는 32비트 카운터

• 8-bit prescaler to enable better control of the interrupt period

인터럽트 주기보다 더 잘 제어할 수 있는 8비트 프리스케일

• Configurable single-shot or auto-reload modes

구성 가능한 싱글(단일)샷 또는 자동 재로드 모드

• Configurable starting values for the counter

카운터에 대해 구성 가능한 시작 값

**8.2.1 Clocking**

All private timers and watchdog timers are always clocked at 1/2 of the CPU frequency (CPU\_3x2x).

모든 개인적인 타이머와 워치독 타이머는 CPU 주파수(CPU\_3x2x)의 2분의 1 클록 된다.

**8.2.2 Interrupt to PS Interrupt Controller**

The interrupts sent to the interrupt controller are described in section 7.2.2 CPU Private Peripheral Interrupts (PPI).

인터럽트 컨트롤러로 전송된 인터럽트들은 섹션7.2.2 CPU Private Peripheral Interrupts (PPI)에 설명되어있다.

**참고 7.2.2 CPU Private Peripheral Interrupts (PPI)**

Each CPU connects to a private set of five peripheral interrupts. The PPIs are listed in Table 7-3.

각 CPU는 5개 주변장치 인터럽트의 개인 세트에 연결된다. PPI은 표 7-3에 나열되어 있다.

The sensitivity types for PPIs are fixed and cannot be changed; therefore, the ICDICFR1 register is read-only, since it specifies the sensitivity types of all the 5 PPIs

PPI의 민감도 유형은 고정되어있으며 변경 할 수 없다. 따라서 ICDICFR1레지스터는 모든 5개PPL의 민감도 유형이 지정되어 있기 때문에 읽기 전용이다.

Note that the fast interrupt (FIQ) signal and the interrupt (IRQ) signal from the PL are inverted and then sent to the interrupt controller.

PL에 인터럽트 신호와 빠른 인터럽트 신호는 반전되고 인터럽트 제어장치에 보내지게 된다.

Therefore, they are active High at the PS-PL interface, although the ICDICFR1 register reflects them as active Low level.

따라서 ICDICFR1레지스터 액티브가 로우 레벨로 반영되더라도 PPI은 PS-PL인터페이스에서 액티브 하이 이다.

**8.2.3 Resets**

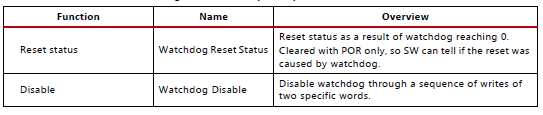
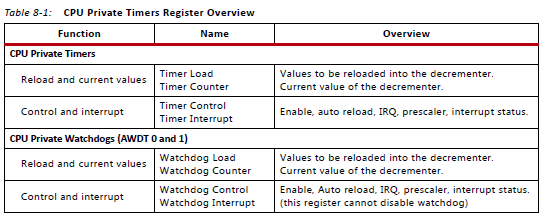
The time and watchdog resets are sent to the PS reset subsystem, see section 26.3 Reset Effects.

타이머와 워치독 리셋은 PS리셋 서브시스템에 전송된다.

**8.2.4 Register Overview**

A register overview of the CPU private and watchdog timers is provided in Table 8-1.

CPU 개인적인 타이머와 워치독 타이머의 레지스터 개요는 표8-1에 제공된다.



Decrementer: 감쇠계: 전자기파의 감폭(진동의 감폭) 을 측정하는 장치

**8.3 Global Timer (GT)**

The Global Timer is fully documented in the *Cortex-A9 MPCore Technical Requirements Document*, sections 4.3 and 4.4 (see Appendix A, Additional Resources).

전역 타이머는 Cortex-A9 MPCore에 완벽하게 설명되어 있습니다. 기술 요구 사항 문서, 섹션 4.3 및 4.4 (부록 A, 추가 리소스 참조).

The global timer is a 64-bit incrementing counter with an auto-incrementing feature.

전역타이머는 자동 증가기능이 있는 64비트 증분카운터이다.

The global timer is memory mapped in the same address space as the private timers.

전역 타이머는 개인적인 타이머와 같은 주소 공간에 매핑되는 메모리다.

The global timer is accessed at reset in secure state only.

전역 타이머는 보안상태에서만 리셋이 액세스된다.

The global timer is accessible to all Cortex-A9 processors.

전역 타이머는 모든 Cortex-A9 프로세서에서 액세스 할 수 있다.

Each Cortex-A9 processor has a 64-bit comparator that is used to assert a private interrupt when the global timer has reached the comparator value.

각각 Cortex-A9 프로세서는 전역 타이머가 비교 값에 도달 할 때 개인 인터럽트를 발생시키는 데 사용되는 64비트 비교기가 있다.

**8.3.1 Clocking**

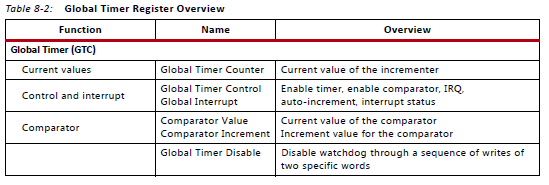
The GTC is always clocked at 1/2 of the CPU frequency (CPU\_3x2x).

GTC(Global Timer Clocking)는 항상 CPU주파수(CPU\_3x2x)의 2분의1로 클록된다.

**8.3.2 Register Overview**

A register overview of the GTC is provided in Table 8-2.

GTC의 레지스터 개요는 표 8-2에 제공된다.



incrementer: 증분기: 전자기파의 증폭(진동의 증폭) 을 측정하는 장치

**8.4 System Watchdog Timer (SWDT)**

In addition to the two CPU private watchdog timers, there is a system watchdog timer (SWDT) for signaling additional catastrophic system failure, such as a PS PLL failure.  
두 개의 CPU 개인용 감시 타이머 외에도 PS PLL 장애와 같은 추가적인 신호(치명적인 시스템 장애) 위한 시스템 워치 독 타이머 (SWDT)가있다

Unlike the AWDT, the SWDT can run off the clock from an external device or the PL, and provides a reset output to an external device or the PL.  
AWDT와 달리 SWDT는 외부 장치 또는 PL에서 클럭을 벗어나 외부 장치 또는 PL에 리셋 출력을 제공합니다.

**8.4.1 Features**

Key features of the available timers/counters are as follows:

사용 가능한 타이머/카운터의 중요 기능

• An internal 24-bit counter

내부 24비트 카운터

• Selectable clock input from:

다음에서 선택 가능한 클록 입력

° Internal PS bus clock (CPU\_1x)

내부 PS버스 클록(CPU\_1x)

° Internal clock (from PL)

내부 클록(PL에서)

° External clock (from MIO)

외부 클록(MIO에서)

• On timeout, outputs one or a combination of:

타임초과시, 다음 중 하나 또는 조합을 출력한다.

° System interrupt (PS)

시스템 인터럽트(PS)

° System reset (PS, PL, MIO)

시스템 리셋(PS, PL, MIO)

• Programmable timeout period:

프로그램 가능한 타임아웃 기간

° Timeout range 32,760 to 68,719,476,736 clock cycles (330 μs to 687.2s at 100 MHz)

타임아웃 범위 32,760~68,719,476,736 클록 사이클(100MHz에서 330 μs ~ 687.2s)

• Programmable output signal duration on timeout:

타임 아웃시 프로그램 가능한 출력 신호 지속기간

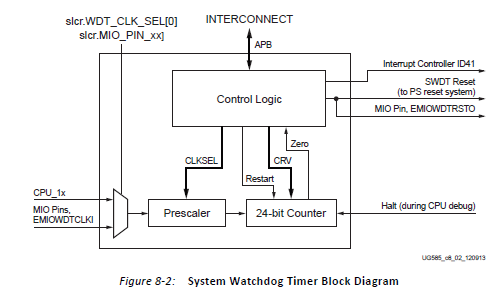
° System interrupt pulse 4, 8, 16, or 32 clock cycles (CPU\_1x clock)

시스템 인터럽트 펄스 4,8,16또는32 클록 사이클 (CPU\_1x clock)

**8.4.2 Block Diagram**

A block diagram of the SWDT is shown in Figure 8-2.

SWD의 블록 다이어 그램을 그림8-2에서 보여준다.



Notes relevant to Figure 8-2:

그림 8-2와 관련된 메모

• SLCR programmable registers (WDT\_CLK\_SEL, MIO control) select the clock input.

• SLCR 프로그래밍 가능 레지스터 (WDT\_CLK\_SEL, MIO 제어)는 클럭 입력을 선택합니다.

• SWDT programmable registers set the values for CLKSEL and CRV.

• SWDT 프로그래머블 레지스터는 CLKSEL 및 CRV에 대한 값을 설정합니다.

• Signal *restart* causes the 24-bit counter to reload the CRV values, and restart counting.

• 신호 재시작으로 24 비트 카운터가 CRV 값을 다시로드하고 카운트를 다시 시작합니다.

• Signal *halt* causes the counter to halt during CPU debug (same behavior as AWDT).

• 신호 정지로 인해 CPU 디버그 중에 카운터가 정지합니다 (AWDT와 동일한 동작).

**8.4.3 Functional Description**

The control logic block has an APB interface connected to the system interconnect.

제어 논리 블록에는 시스템 상호 연결에 연결된 APB 인터페이스가 있습니다.

Each write data received from the APB has a key field which must match the key of the register in order to be able to write to the register.

APB로부터 수신 된 각각의 기록 데이터는 레지스터에 기록 할 수 있도록 레지스터의 키와 일치 해야 하는 키 필드를 갖는다.

The Zero Mode register controls the behavior of the SWDT when its internal 24-bit counter reaches zero.

제로 모드 레지스터는 내부 24 비트 카운터가 0에 도달 할 때 SWDT의 동작을 제어합니다.

Upon receiving a *zero* signal, the control logic block asserts the interrupt output signal for IRQLN clock cycles if both WDEN and IRQEN are set, and also asserts the reset output signals for approximately one CPU\_1x cycle if WDEN is set.

제로 신호를 받으면 WDEN 와 IRQEN이 모두 설정되면 IRQLN 클럭 사이클에 대한 인터럽트 출력 신호를 보내고 WDEN이 설정된 경우 약 1 CPU\_1x주기 동안 리셋 출력 신호를 보낸다.

The 24-bit counter then stays at zero until it is restarted.

24 비트 카운터는 다시 시작될 때까지 0으로 유지됩니다.

The Counter Control register sets the timeout period, by setting reload values in swdt.CONTROL[CLKSET] and swdt.CONTROL[CRV] to control the prescaler and the 24-bit counter.

카운터 제어 레지스터는 프리스케일러와 24 비트 카운터를 제어하기 위해 swdt.CONTROL [CLKSET] 와 swdt.CONTROL [CRV]에 재설정 값을 설정하여 제한 시간을 설정합니다.

The Restart register is used to restart the counting process.

다시 시작 레지스터는 계산 프로세스를 다시 시작하는 데 사용됩니다.

Writing to this register with a matched key causes the prescaler and the 24-bit counter to reload the values from CRV signals.

이 레지스터에 일치하는 키를 쓰면 프리스케일러와 24 비트 카운터가 CRV 신호의 값을 다시 로드 합니다.

The Status register shows whether the 24-bit counter reaches zero.

Status 레지스터는 24 비트 카운터가 0인지 여부를 보여줍니다.

Regardless of the WDEN bit in the Zero Mode register, the 24-bit counter always keeps counting down to zero if it is not zero and the selected clock source is present.

제로 모드 레지스터의 WDEN 비트와 관계없이 24 비트 카운터는 0이 아니고 선택한 클럭 소스가 있는 경우 항상 0으로 카운트 다운을 유지합니다.

Once it reaches zero, the WDZ bit of the Status register is set and remains set until the 24-bit counter is restarted.

일단 0에 도달하면 Status 레지스터의 WDZ 비트가 설정되고 24 비트 카운터가 다시 시작될 때까지 설정 상태를 유지합니다.

The prescaler block divides down the selected clock input.

prescaler 블록은 선택된 클럭 입력을 나눕니다.

The CLKSEL signal is sampled at every rising clock edge.

CLKSEL 신호는 모든 상승 클록 에지에서 샘플링됩니다

The internal 24-bit counter counts down to zero and stays at zero until it is restarted.

내부 24 비트 카운터는 0으로 카운트 다운하고 다시 시작할 때까지 0으로 유지됩니다.

While the counter is at zero, the zero output signal is High.

카운터가 0에있는 동안, 제로 출력 신호는 High다.

**Interrupt to PS Interrupt Controller**

PS 인터럽트 컨트롤러에 인터럽트

The pulse length from the SWDT (four CPU\_1x clock cycles) is sufficient for the interrupt controller to capture the interrupt using rising-edge sensitivity.

SWDT로부터의 펄스 길이 (4 개의 CPU\_1x 클럭 사이클)는 인터럽트 컨트롤러가 상승 에지 감도를 사용하여 인터럽트를 캡처하는 데 충분합니다.

**Reset**

The watchdog reset is sent to the PS reset subsystem to cause a non-POR reset, see section 26.3 Reset Effects.

워치 독 리셋은 PS 리셋 서브 시스템으로 전송되어 non-POR 리셋이 발생합니다. 26.3 절을 참조하십시오

The reset output to the MIO pin or EMIOWDTRSTO is active High.

MIO 핀 또는 EMIOWDTRSTO에 대한 리셋 출력은 High이다.

**TIP:** *To generate a signal pulse for the PS\_POR\_B and other board resets, route the EMIOWDTRSTO signal from the SWDT through the PL and to a pin that can be externally latched to generate a valid reset pulse.*

PS\_POR\_B 와 다른 보드 리셋을 위한 신호 펄스를 생성하려면 SWDT에서 PLOW를 통해 EMIOWDTRSTO 신호를 외부 적으로 latch하여 유효한 리셋 펄스를 생성 할 수 있는 핀에 연결하십시오.

Latch : **Latch**는 하나의 데이터 입력, 하나의 클럭 입력 그리고 하나의 출력을 갖는다.

*Alternatively, use an external watchdog timer device that is managed by PS software via a GPIO output pin.*

또는 GPIO 출력 핀을 통해 PS 소프트웨어로 관리되는 외부 워치 독 타이머 장치를 사용하십시오.

*The PS\_POR\_B reset pulse width requirements are defined in the data sheet.*

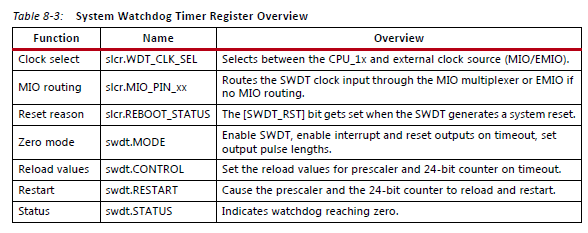
PS\_POR\_B 리셋 펄스 폭 요구 사항은 데이터 시트에 정의되어 있습니다.

**8.4.4 Register Overview**

8.4.4 레지스터 개요

A register overview of the SWDT is provided in Table 8-3.

표 8-3에는 SWDT에 대한 레지스터 개요가 나와 있습니다

**

**8.4.5 Programming Model**

8.4.5 프로그래밍 모델

**System Watchdog Timer Enable Sequence**

시스템 워치 독 타이머 활성화 시퀀스

1. Select clock input source using the slcr.WDT\_CLK\_SEL[SEL] bit:

Ensure that the SWDT is disabled (swdt.MODE[WDEN] = 0) and the clock input source to be selected is running before proceeding with this step.

1. slcr.WDT\_CLK\_SEL [SEL] 비트를 사용하여 클럭 입력 소스를 선택하십시오 :

이 단계를 진행하기 전에 SWDT가 비활성화되어 있는지 확인하고 (swdt.MODE [WDEN] = 0) 선택한 클럭 입력 소스가 실행 중인지 확인하십시오.

Changing the clock input source when the SWDT is enabled results in unpredictable behavior.

SWDT가 활성화되어있을 때 클록 입력 소스를 변경하면 예기치 않은 동작이 발생합니다.

Changing the clock input source to a non-running clock results in APB access hang.

클럭 입력 소스를 비작동 클럭으로 변경하면 APB 액세스가 중지됩니다.

2. Set the timeout period (Counter Control register):

The swdt.CONTROL[CKEY] field must be 0x248 to be able to write this register.

2. 타임 아웃주기 설정 (카운터 제어 레지스터) :

이 레지스터를 작성하려면 swdt.CONTROL [CKEY] 필드가 0x248이어야합니다.

3. Enable the counter; enable output pulses; set up output pulse lengths (Zero Mode register):

The swdt.MODE[ZKEY] field must be 0xABC to be able to write this register.

3. 카운터를 활성화하십시오. 출력 펄스를 활성화한다. 출력 펄스 길이 설정 (제로 모드 레지스터) :

이 레지스터를 쓸 수 있으려면 swdt.MODE [ZKEY] 필드가 0xABC 여야합니다.

Ensure that IRQLN meets the specified minimum values.

IRQLN이 지정된 최소값을 충족하는지 확인하십시오.

4. To run the SWDT with a different setting, disable the timer first (swdt.MODE[ZKEY] bit).

Then repeat steps 1, 2, and 3.

4. 다른 설정으로 SWDT를 실행하려면 먼저 타이머를 비활성화하십시오 (swdt.MODE [ZKEY] 비트).

그런 다음 1, 2 및 3 단계를 반복하십시오.

**8.4.6 Clock Input Option for SWDT**

SWDT를위한 8.4.6 클록 입력 옵션

The following code shows how the AP SoC selects the clock source for SWDT:

다음 코드는 AP SoC가 SWDT에 대한 클럭 소스를 선택하는 방법을 보여준다.

if slcr.WDT\_CLK\_SEL[0] is 0, use CPU\_1X

slcr.WDT\_CLK\_SEL [0]이 0이면 CPU\_1X

else if slcr.MIO\_PIN\_14[7:0] is 01100000, use MIO pin 14

그렇지 않으면 slcr.MIO\_PIN\_14 [7 : 0]이 01100000이면 MIO 핀 14를 사용하십시오

else if slcr.MIO\_PIN\_26[7:0] is 01100000, use MIO pin 26

그렇지 않으면 slcr.MIO\_PIN\_26 [7 : 0]이 01100000이면 MIO 핀 26을 사용하십시오

else if slcr.MIO\_PIN\_38[7:0] is 01100000, use MIO pin 38

그렇지 않으면 slcr.MIO\_PIN\_38 [7 : 0]이 01100000이면 MIO 핀 38을 사용하십시오

else if slcr.MIO\_PIN\_50[7:0] is 01100000, use MIO pin 50

그렇지 않으면 slcr.MIO\_PIN\_50 [7 : 0]이 01100000이고 MIO 핀 50을 사용하십시오

else if slcr.MIO\_PIN\_52[7:0] is 01100000, use MIO pin 52

그렇지 않으면 slcr.MIO\_PIN\_52 [7 : 0]이 01100000이면 MIO 핀 52를 사용하십시오

else use EMIOWDTCLKI

그렇지 않으면 EMIOWDTCLKI를 사용하십시오.

**8.4.7 Reset Output Option for SWDT**

SWDT에 대한 출력 옵션 재설정

The following code shows how the AP SoC selects the reset output pin for SWDT:

다음 코드는 AP SoC가 SWDT에 대한 리셋 출력 핀을 선택하는 방법을 보여줍니다.

if slcr.WDT\_CLK\_sel[0] is 0, no output (to PS reset system only)

slcr.WDT\_CLK\_sel [0]이 0이면 출력이 없습니다 (PS 재설정 시스템에만 해당)

else if slcr.MIO\_PIN\_15[7:0] is 01100000, use MIO pin 15

그렇지 않으면 slcr.MIO\_PIN\_15 [7 : 0]이 01100000이고 MIO 핀 15를 사용하십시오

else if slcr.MIO\_PIN\_27[7:0] is 01100000, use MIO pin 27

그렇지 않으면 slcr.MIO\_PIN\_27 [7 : 0]이 01100000이면 MIO 핀 27을 사용하십시오

else if slcr.MIO\_PIN\_39[7:0] is 01100000, use MIO pin 39

그렇지 않으면 slcr.MIO\_PIN\_39 [7 : 0]이 01100000이면 MIO 핀 39를 사용하십시오

else if slcr.MIO\_PIN\_51[7:0] is 01100000, use MIO pin 51

그렇지 않은 경우 slcr.MIO\_PIN\_51 [7 : 0]이 01100000이면 MIO 핀 51

else if slcr.MIO\_PIN\_53[7:0] is 01100000, use MIO pin 53

그렇지 않으면 slcr.MIO\_PIN\_53 [7 : 0]이 01100000이면 MIO 핀 53을 사용하십시오

else use EMIOWDTRSTO

그렇지 않으면 EMIOWDSTREST를 사용하십시오.

**8.5 Triple Timer Counters (TTC)**

8.5 트리플 타이머 카운터 (TTC)

The TTC contains three independent timers/counters.

TTC에는 세 개의 독립적 인 타이머 / 카운터가 있습니다.

There are two TTC modules in the PS, for a total of six timers/counters.

PS에는 두 개의 TTC 모듈이 있으며 총 6 개의 타이머 / 카운터가 있습니다

TTC 1 controller can be configured for secure or non-secure mode using the nic301\_addr\_region\_ctrl\_registers.security\_apb [ttc1\_apb] register bit.

TTC 1 컨트롤러는 nic301\_addr\_region\_ctrl\_register.security\_apb [ttc1\_apb] 레지스터 비트를 사용하여 안전 모드 또는 비보안 모드로 구성 할 수 있습니다.

The three timers within a TTC controller have the same security state.

TTC 컨트롤러 내의 3 개의 타이머는 동일한 보안 상태를 가집니다.

**8.5.1 Features**

8.5.1 특징

Each of the triple timer counters has:

트리플 타이머 카운터는 다음과 같습니다.

• Three independent 16-bit prescalers and 16-bit up/down counters

• 3 개의 독립적 인 16 비트 프리스케일러 및 16 비트 업 / 다운 카운터

• Selectable clock input from:

• 다음에서 선택 가능한 클럭 입력 :

° Internal PS bus clock (CPU\_1x)

° 내부 PS 버스 클럭 (CPU\_1x)

° Internal clock (from PL)

° 내부 클럭 (PL에서)

° External clock (from MIO)

° 외부 클럭 (MIO에서)

• Three interrupts, one for each counter

• 세 개의 인터럽트, 각 카운터 당 하나

• Interrupt on overflow, at regular interval, or counter matching programmable values

• 오버플로, 일정한 간격 또는 카운터 매칭 가능 프로그램 값에 대한 인터럽트

• Generates waveform output (for example, PWM) through the MIO and to the PL

• MIO 및 PL을 통해 파형 출력 (예 : PWM) 생성

**8.5.2 Block Diagram**

8.5.2 블록 다이어그램

A block diagram of the TTC is shown in Figure 8-3.

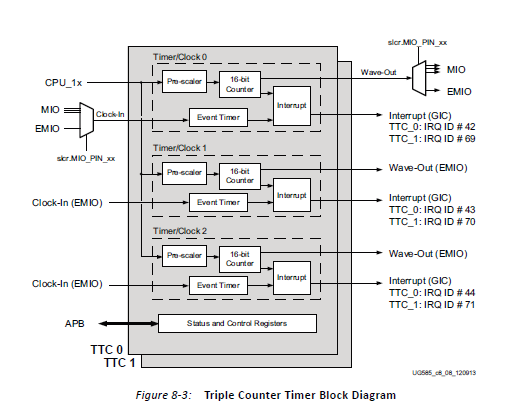
그림 8-3에는 TTC의 블록 다이어그램이 나와 있습니다.

The clock-in and wave-out multiplexing for Timer/Clock 0 is controlled by the slcr.MIO\_PIN\_xx registers.

타이머 / 클럭 0에 대한 클럭 in 과 웨이브 아웃 멀티플렉싱은 slcr.MIO\_PIN\_xx 레지스터에 의해 제어됩니다.

If no selection is made in these registers, then the default becomes the EMIO interface.

이러한 레지스터에서 선택이 없으면 기본값은 EMIO 인터페이스가 됩니다.



**8.5.3 Functional Description**

8.5.3 기능 설명

Each prescaler module can be independently programmed to use the PS internal bus clock (CPU\_1x), or an external clock from the MIO or the PL.

각 프리스케일러 모듈은 PS 내부 버스 클럭 (CPU\_1x) 또는 MIO 또는 PL의 외부 클록을 사용하도록 독립적으로 프로그래밍 할 수 있습니다.

For an external clock, SLCR registers determine the exact pinout through the MIO or from the PL.

외부 클록의 경우 SLCR 레지스터는 MIO 또는 PL을 통해 정확한 핀아웃을 결정합니다.

The selected clock is then divided down from /2 to /65536, before being applied to the counter.

선택한 클록은 카운터에 적용되기 전에 / 2에서 / 65536으로 분할됩니다.

The counter module can count up or count down, and can be configured to count for a given interval.

카운터 모듈은 카운트 업 또는 카운트 다운 할 수 있으며 주어진 간격 동안 카운트하도록 구성 할 수 있습니다.

It also compares three match registers to the counter value, and generate an interrupt if one matches.

또한 세 개의 일치 레지스터를 카운터 값과 비교하여 일치하는 항목이 하나 있는 경우 인터럽트를 생성합니다.

The interrupt module combines interrupts of various types: counter interval, counter matches, counter overflow, event timer overflow.

인터럽트 모듈은 카운터 간격, 카운터 일치, 카운터 오버플로, 이벤트 타이머 오버플로 등 다양한 유형의 인터럽트를 결합합니다.

Each type can be individually enabled.

각 유형은 개별적으로 사용할 수 있습니다.

**Modes of Operation**

작동 모드

Each counter module can be independently programmed to operate in either of the following two modes:

작동 모드 각 카운터 모듈은 다음 두 가지 모드 중 하나에서 작동하도록 독립적으로 프로그래밍 할 수 있습니다.

**Interval mode:** The counter increments or decrements continuously between 0 and the value of the Interval register, with the direction of counting determined by the DEC bit of the Counter Control register.

인터벌 모드 : 카운터는 카운터 제어 레지스터의 DEC 비트에 의해 결정된 카운팅 방향으로 0과 인터벌 레지스터 값 사이에서 연속적으로 증가 또는 감소합니다.

An interval interrupt is generated when the counter passes through zero. The corresponding match interrupt is generated when the counter value equals one of the Match registers.

간격 인터럽트는 카운터가 0을 통과 할 때 생성됩니다. 일치하는 일치 인터럽트는 카운터 값이 일치 레지스터 중 하나와 같을 때 생성됩니다

**Overflow mode:** The counter increments or decrements continuously between 0 and 0xFFFF, with the direction of counting determined by the DEC bit of the Counter Control register.

오버플로 모드 : 카운터 제어 레지스터의 DEC 비트에 의해 결정된 카운팅 방향으로 0과 0xFFFF 사이에서 카운터가 연속적으로 증가 또는 감소합니다.

An overflow interrupt is generated when the counter passes through zero.

오버플로 인터럽트는 카운터가 0을 통과 할 때 생성됩니다.

The corresponding match interrupt is generated when the counter value equals one of the Match registers.

일치하는 일치 인터럽트는 카운터 값이 일치 레지스터 중 하나와 같을 때 생성됩니다.

**Event Timer Operation**

이벤트 타이머 작동

The event timer operates by having an internal (invisible to users) 16-bit counter clocked at CPU\_1x which:

이벤트 타이머는 내부 (사용자에게는 보이지 않는) CPU\_1x에서 16 비트 카운터가 클록 됨으로써 작동합니다.

• Resets to 0 during the non-counting phase of the external pulse

• 외부 펄스가 카운트되지 않는 단계에서 0으로 리셋

• Increments during the counting phase of the external pulse

The Event Control Timer register controls the behavior of the internal counter:

• 외부 펄스의 카운팅 단계 동안 증가

이벤트 제어 타이머 레지스터는 내부 카운터의 동작을 제어합니다.

• **E\_En bit:** When 0, immediately resets the internal counter to 0, and stops incrementing

• E\_En 비트 : 0이면 즉시 내부 카운터를 0으로 재설정하고 증가를 중지합니다

• **E\_Lo bit:** Specifies the counting phase of the external pulse

• E\_Lo 비트 : 외부 펄스의 카운팅 위상을 지정합니다

• **E\_Ov bit:** Specifies how to handle overflow at the internal counter (during the counting phase of the external pulse)

• E\_Ov 비트 : 내부 카운터에서 오버 플로우를 처리하는 방법을 지정합니다 (외부 펄스의 카운팅 단계 중에).

° When 0: Overflow causes E\_En to be 0 (see E\_En bit description)

° 0 일 때 : 오버플로는 E\_En을 0으로 만듭니다 (E\_En 비트 설명 참조)

° When 1: Overflow causes the internal counter to wrap around and continues incrementing

° 1 일 때 : 오버플로가 발생하면 내부 카운터가 랩되고 증가를 계속합니다

° An interrupt is always generated (subject to further enabling through another register) when an overflow occurs.

° 오버플로가 발생할 때 항상 인터럽트가 생성됩니다 (다른 레지스터를 통해 추가로 활성화해야 함).

The Event register is updated with the non-zero value of the internal counter at the end of the counting-phase of the external pulse; therefore, it shows the widths of the external pulse, measured in number of cycles of CPU\_1x.

이벤트 레지스터는 외부 펄스의 counting-phase가 끝날 때 내부 카운터의 0이 아닌 값으로 업데이트됩니다. 따라서 외부 펄스의 폭을 CPU\_1x의 사이클 수로 측정합니다.

If the internal counter is reset to 0, due to overflow, during the counting phase of the external pulse, the Event register will not be updated and maintains the old value from the last non-overflowing counting operation.

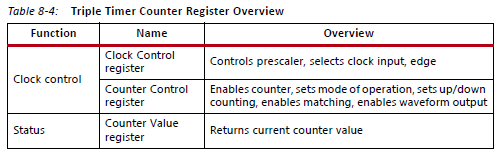
오버플로로 인해 내부 카운터가 0으로 재설정 된 경우 외부 펄스의 카운팅 단계에서 Event 레지스터가 업데이트되지 않고 마지막 오버플로가 아닌 카운팅 작업에서 이전 값을 유지합니다

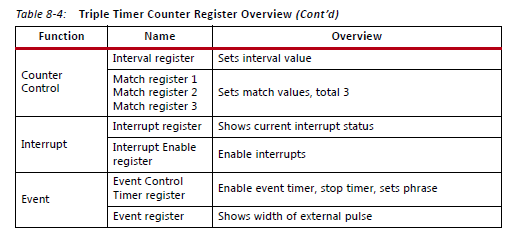
**8.5.4 Register Overview**

8.5.4 레지스터 개요

A register overview of the TTC is provided in Table 8-4.

표 8-4에는 TTC의 레지스터 개요가 나와 있습니다.





**8.5.5 Programming Model**

8.5.5 프로그래밍 모델

**Counter Enable Sequence**

카운터 활성화 시퀀스

1. Select clock input source, set prescaler value (slcr.MIO\_MUX\_SEL registers, TTC Clock Control register).

1. 클럭 입력 소스를 선택하고 프리스케일러 값을 설정하십시오 (slcr.MIO\_MUX\_SEL 레지스터, TTC 클록 제어 레지스터).

Ensure TTC is disabled (ttc.Counter\_Control\_x [DIS] = 1) before proceeding with this step.

이 단계를 진행하기 전에 TTC가 비활성화되어 있는지 확인하십시오 (ttc.Counter\_Control\_x [DIS] = 1).

2. Set interval value (Interval register). This step is optional, for interval mode only.

2. 간격 값 (간격 레지스터)을 설정하십시오. 이 단계는 간격 모드에 대해서만 선택 사항입니다.

3. Set match value (Match registers). This step is optional, if matching is to be enabled.

3. 일치 값 (일치 레지스터)을 설정하십시오. 일치를 사용하려면이 단계는 선택 사항입니다.

4. Enable interrupt (Interrupt Enable register). This step is optional, if interrupt is to be enabled.

4. 인터럽트 (인터럽트 활성화 레지스터)를 활성화합니다. 인터럽트를 사용하려면이 단계를 선택하십시오.

5. Enable/disable waveform output, enable/disable matching, set counting direction, set mode, enable counter (TTC Counter Control register).

5. 파형 출력을 활성화 / 비활성화, 일치 여부를 설정, 카운트 방향 설정, 모드 설정, 카운터 활성화 (TTC 카운터 제어 레지스터).

This step starts the counter.

이 단계는 카운터를 시작합니다.

**Counter Stop Sequence**

카운터 정지 시퀀스

1. Read back the value of the Counter Control register.

1. 카운터 제어 레지스터의 값을 다시 읽습니다.

2. Set DIS bit to 1, while keeping other bits.

2. 다른 비트를 유지하면서 DIS 비트를 1로 설정하십시오.

3. Write back to Counter Control register.

3. 카운터 제어 레지스터에 다시 기록하십시오.

**Counter Restart Sequence**

카운터 재시작 순서

1. Read back the value of Counter Control register.

1. Counter Control 레지스터의 값을 다시 읽습니다.

2. Set RST bit to 1, while keeping other bits.

2. 다른 비트를 유지하면서 RST 비트를 1로 설정하십시오.

3. Write back to Counter Control register.

3. 카운터 제어 레지스터에 다시 기록하십시오.

**Event Timer Enable Sequence**

이벤트 타이머 활성화 시퀀스

1. Select external pulse source (slcr.MIO\_MUX\_SEL registers).

1. 외부 펄스 소스 (slcr.MIO\_MUX\_SEL 레지스터)를 선택하십시오.

The width of the selected external pulse is measured in CPU\_1x period.

선택된 외부 펄스의 폭은 CPU\_1x주기에서 측정됩니다.

2. Set overflow handling, select external pulse level, enable the event timer (Event Control Timer register).

2. 오버 플로우 처리 설정, 외부 펄스 레벨 선택, 이벤트 타이머 사용 가능 (이벤트 제어 타이머 레지스터).

This step starts measuring the width of the selected level (High or Low) of the external pulse.

이 단계는 외부 펄스의 선택된 레벨 (High 또는 Low)의 폭 측정을 시작합니다.

3. Enable interrupt (Interrupt Enable register).

3. 인터럽트 (인터럽트 활성화 레지스터)를 활성화합니다.

This step is optional, if interrupt is to be enabled.

인터럽트를 사용하려면이 단계를 선택하십시오.

4. Read the measured width (Event register). Note that the returned value is not correct when overflow happened.

4. 측정 된 너비 (이벤트 레지스터)를 읽습니다. 오버플로가 발생하면 반환 된 값이 올바르지 않습니다.

See the description for the E\_Ov bit of the Event Control Timer register in section 8.5.3 Functional Description.

8.5.3 기능 설명의 이벤트 제어 타이머 레지스터의 E\_Ov 비트에 대한 설명을 참조하십시오.

**Interrupt Clear and Acknowledge Sequence**

인터럽트 클리어 및 승인 시퀀스

1. Read Interrupt register: All bits in the Interrupt register are cleared on read.

1. 인터럽트 레지스터 읽기 : 인터럽트 레지스터의 모든 비트는 읽기시 지워집니다.

**8.5.6 Clock Input Option for Counter/Timer**

8.5.6 카운터 / 타이머를위한 클록 입력 옵션

The following shows how AP SoC selects the clock source for TTC0 counter/timer 0:

다음은 AP SoC가 TTC0 카운터 / 타이머 0에 대한 클럭 소스를 선택하는 방법을 보여줍니다

if slcr.MIO\_PIN\_19[6:0] is 1100000, use MIO pin 19

else if slcr.MIO\_PIN\_31[6:0] is 1100000, use MIO pin 31

else if slcr.MIO\_PIN\_43[6:0] is 1100000, use MIO pin 43

else use EMIOTTC0CLKI0

TTC0 counter/timer 1 can use only EMIOTTC0CLKI1.

TTC0 counter/timer 2 can use only EMIOTTC0CLKI2.

The following shows how Zynq SoC selects the clock source for TTC1 counter/timer 0:

if slcr.MIO\_PIN\_17[6:0] is 1100000, use MIO pin 17

else if slcr.MIO\_PIN\_29[6:0] is 1100000, use MIO pin 29

else if slcr.MIO\_PIN\_41[6:0] is 1100000, use MIO pin 41

else use EMIOTTC1CLKI0

TTC1 counter/timer 1 can use only EMIOTTC1CLKI1.

TTC1 counter/timer 2 can use only EMIOTTC1CLKI2.

**8.6 I/O Signals**

8.6 I / O 신호

Timer I/O signals are identified in Table 8-5. The MIO pins and any restrictions based on device version are shown in the MIO table in section 2.5.4 MIO-at-a-Glance Table.

타이머 I / O 신호는 표 8-5에 나와 있습니다. 디바이스 버전에 기반한 MIO 핀과 제한 사항은 2.5.4 MIO-at-a-Glance 표 섹션의 MIO 테이블에 나와 있습니다.

There are two triple timer counters (TTC0 and TTC1) in the system.

시스템에는 두 개의 트리플 타이머 카운터 (TTC0 및 TTC1)가 있습니다.

Each TTC has three sets of interface signals: clock in and wave out for counter/timers 0, 1, and 2.

각 TTC에는 카운터 / 타이머 0, 1 및 2에 대한 클록 입력 및 웨이브 아웃의 세 가지 인터페이스 신호 세트가 있습니다.

For each triple timer counter, the signals for counter/timer 0 can be routed to the MIO using the MIO\_PIN registers.

각 트리플 타이머 카운터에 대해 카운터 / 타이머 0 신호는 MIO\_PIN 레지스터를 사용하여 MIO로 라우팅 될 수 있습니다

If the clock in or wave out signal is not selected by the MIO\_PIN register, then the signal is routed to EMIO by default.

클록 인 또는 웨이브 아웃 신호가 MIO\_PIN 레지스터에 의해 선택되지 않으면, 신호는 기본적으로 EMIO로 라우팅됩니다.

The signals for counter/timers 1 and 2 are only available through the EMIO.

카운터 / 타이머 1 및 2의 신호는 EMIO를 통해서만 사용할 수 있습니다.

