

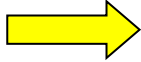
Computer Engineering WS 2012

Timer/Counter

HTM – SHF - SWR



Übersicht



- ▶ **Einleitung**
- ▶ **Output-Compare Funktion**
- ▶ **Input-Capture Funktion.**
- ▶ **Mehrfachzugriff auf Counter-Register.**
- ▶ **Timer/Counter im LPC2468**

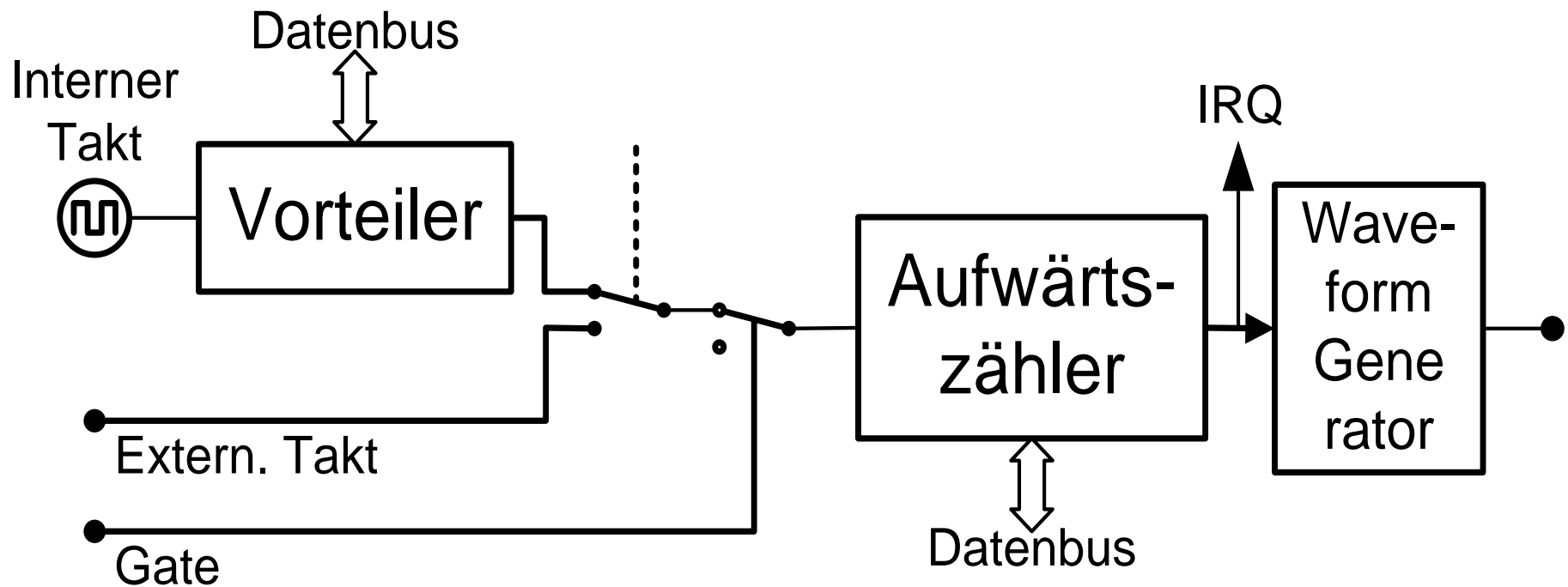


Aufgaben

- ▶ Erzeugen von Zeitfunktionen
 - ▶ Programmteile, die zu genau vorgegebenen Zeitpunkten ausgeführt werden sollen.
 - ▶ Externe Signale zur Steuerung von Peripheriegeräten.
- ▶ Messung von Zeitfunktionen
 - ▶ Dauer von externen Signalen.
 - ▶ Frequenz von externen Signalen.
- ▶ Zählen von Ereignissen.



Prinzipieller Aufbau





Anwendungen: Zeitgeber

- ▶ Periodisch oder Single Shot
- ▶ Interner Takt
- ▶ Beispielanwendungen
 - ▶ Systemtakt für Betriebssysteme
 - ▶ Schrittmotorsteuerung

Zeitmessung

- ▶ Interner Takt
- ▶ Start und/oder Stop durch externes Ereignis
- ▶ Beispielanwendungen
 - ▶ Bestimmung der Baudrate

Frequenzmessung

- ▶ Externer Takt
- ▶ Start und Stop des Zählers per Software

Zähler

- ▶ Externe Ereignisse werden als Takt verwendet
- ▶ Beispielanwendungen
 - ▶ Auswertung von Drehimpulsgebern



Überwachung von und Reaktion auf Timer-Ereignisse

▶ Polling

▶ Kontinuierliche Abfrage

- des Zählerstandes oder
- des Statusregisters:

```
/* warte, bis Timer-Ereignis auftritt */  
  
while(( TIMER0_IR & MR0) == 0 ){  
}
```

▶ Interrupt

▶ Automatische Änderung von Signalen an Anschlusspins

▶ Mögliche Aktionen an Ausgängen bei Timer-Ereignissen:

- Setzen
- Löschen
- Toggeln

Zeitliche Auflösung

- ▶ bestimmt durch verwendete
 - ▶ Taktfrequenz

$$\Delta T = \frac{N_{Vorteiler}}{f_{Takt}}$$

Intervall

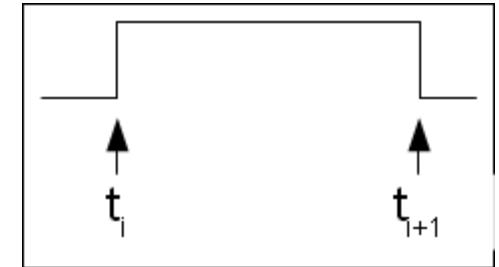
- ▶ bestimmt durch verwendete
 - ▶ Taktfrequenz
 - ▶ Anzahl Bits des Zählers

$$T_{Max} = 2^N \cdot \frac{N_{Vorteiler}}{f_{Takt}}$$



Übung: Dimensionierung eines Timers

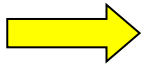
- ▶ Mit Hilfe eines 16-Bit Timers soll eine Impulsdauer zwischen 0.5 und 2.0 Sekunden gemessen werden. Die Systemfrequenz beträgt 8 MHz. Als Vorteiler stehen die Werte 1, 8, 64, 256 und 1024 zur Verfügung.



1. Wie muss der Vorteiler eingestellt werden?
2. Wie genau kann die Zeit erfasst werden?
Verwenden Sie bitte handhabbare Einheiten.

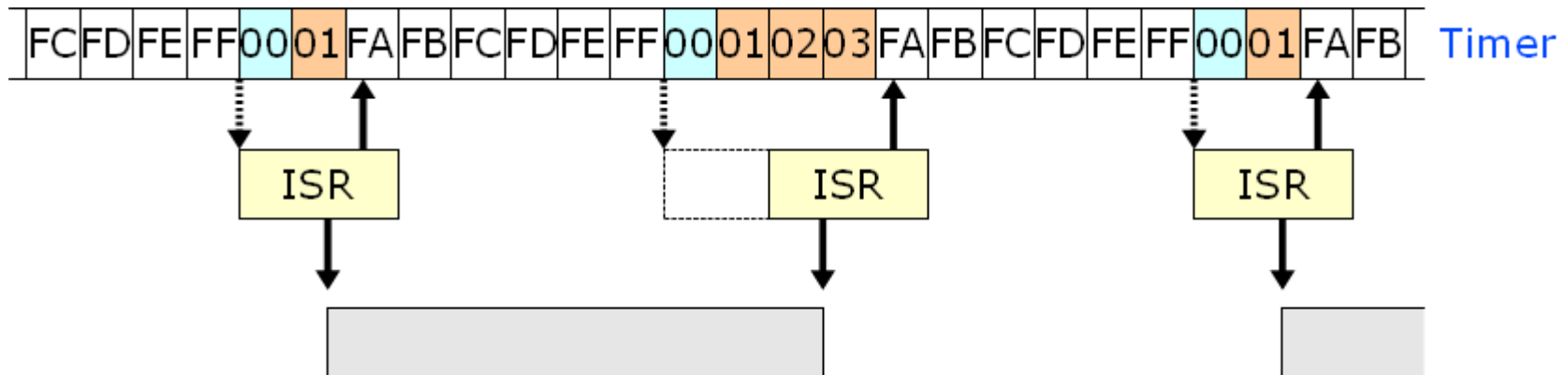


Übersicht



- ▶ Einleitung
- ▶ Output-Compare Funktion
- ▶ Input-Capture Funktion.
- ▶ Mehrfachzugriff auf Counter-Register.
- ▶ Timer/Counter im LPC2468

Erzeugung eines periodischen Ausgangssignals



▶ Ablauf:

- ▶ Beim Überlauf des Zählers wird Overflow-Interrupt ausgelöst.
- ▶ In der ISR wird der Zähler auf den Ausgangswert zurückgesetzt.

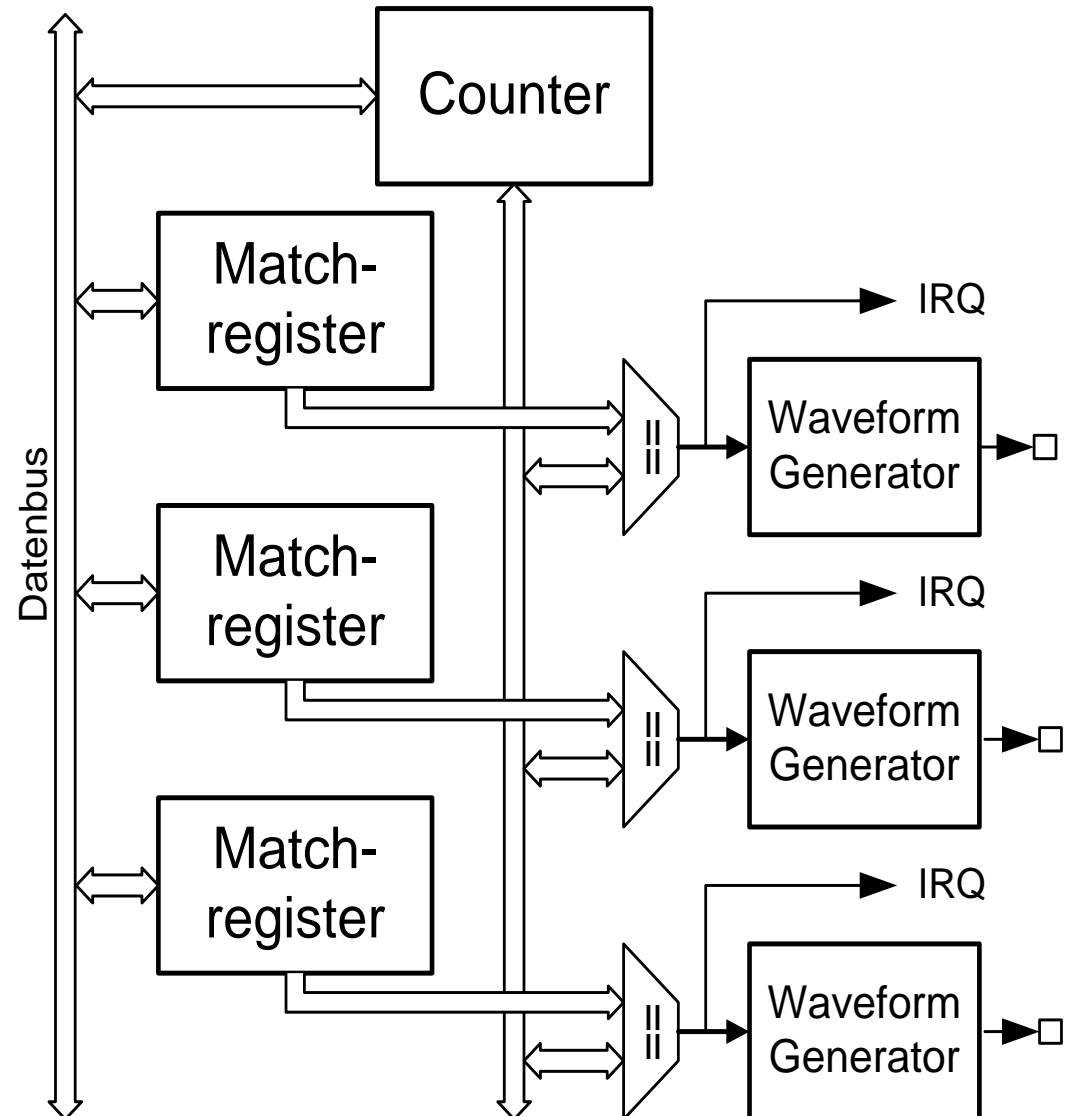
▶ Probleme:

- ▶ Latenz- und Bearbeitungszeit des Interrupts bewirken:
 - Falsche Frequenz
 - Jitter.
- ▶ Timer kann nur ein Signal zur Zeit erzeugen.

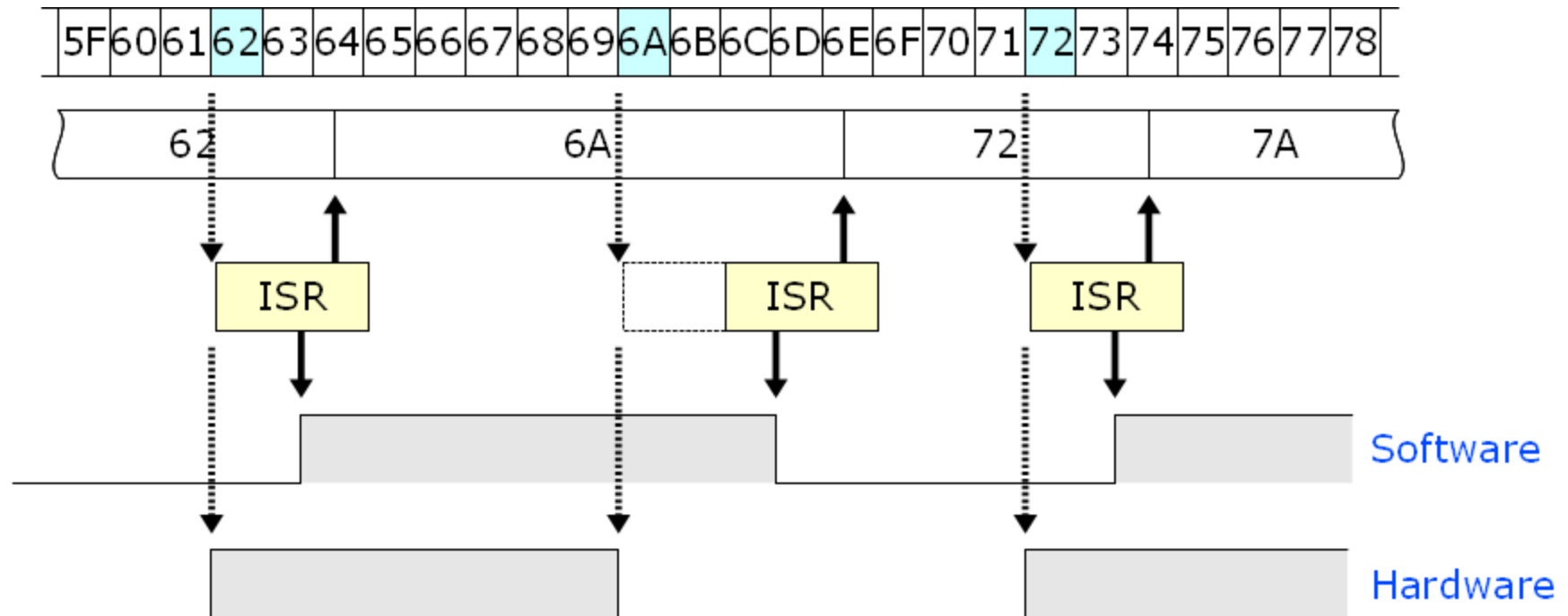


Timer mit Output-Compare-Funktion

- Ermöglicht die Erzeugung von mehreren unabhängigen Signalen



Output-Compare-Funktion: Periodisches Ausgangssignal

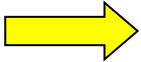


- ▶ **Hardwaregesteuerter Ausgang:**
 - ▶ **taktgenau**
- ▶ **Softwaregesteuerter Ausgang:**
 - ▶ **richtige Frequenz**
 - ▶ **aber Jitter**

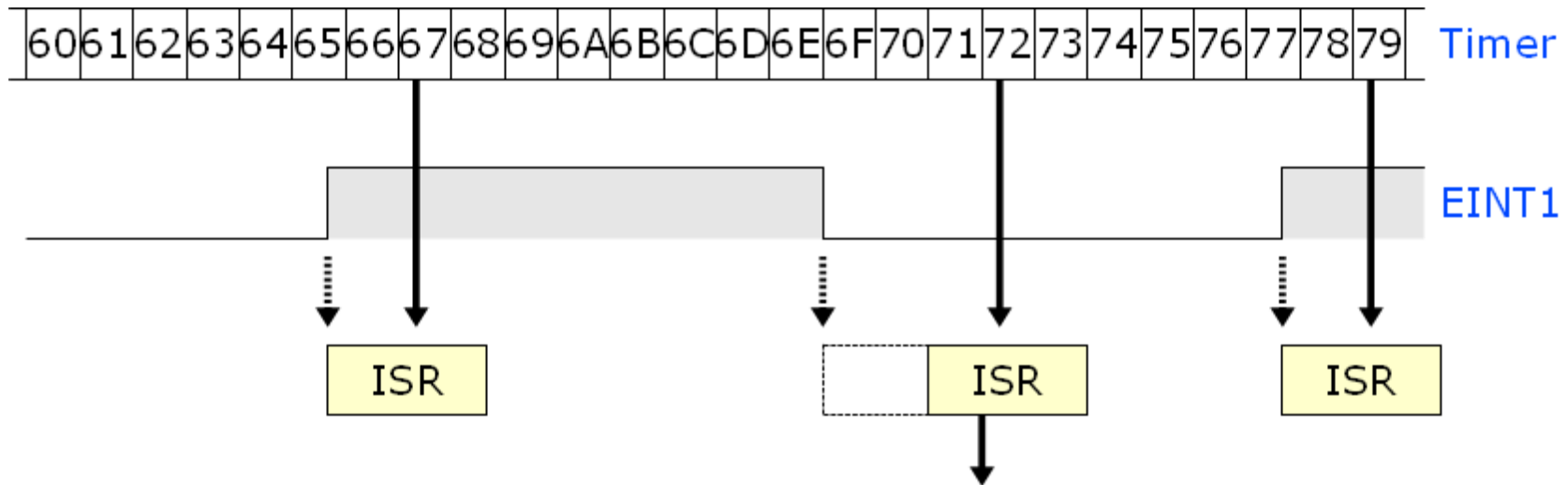


Übersicht

- ▶ Einleitung
- ▶ Output-Compare Funktion
- ▶ Input-Capture Funktion.
- ▶ Mehrfachzugriff auf Counter-Register.
- ▶ Timer/Counter im LPC2468



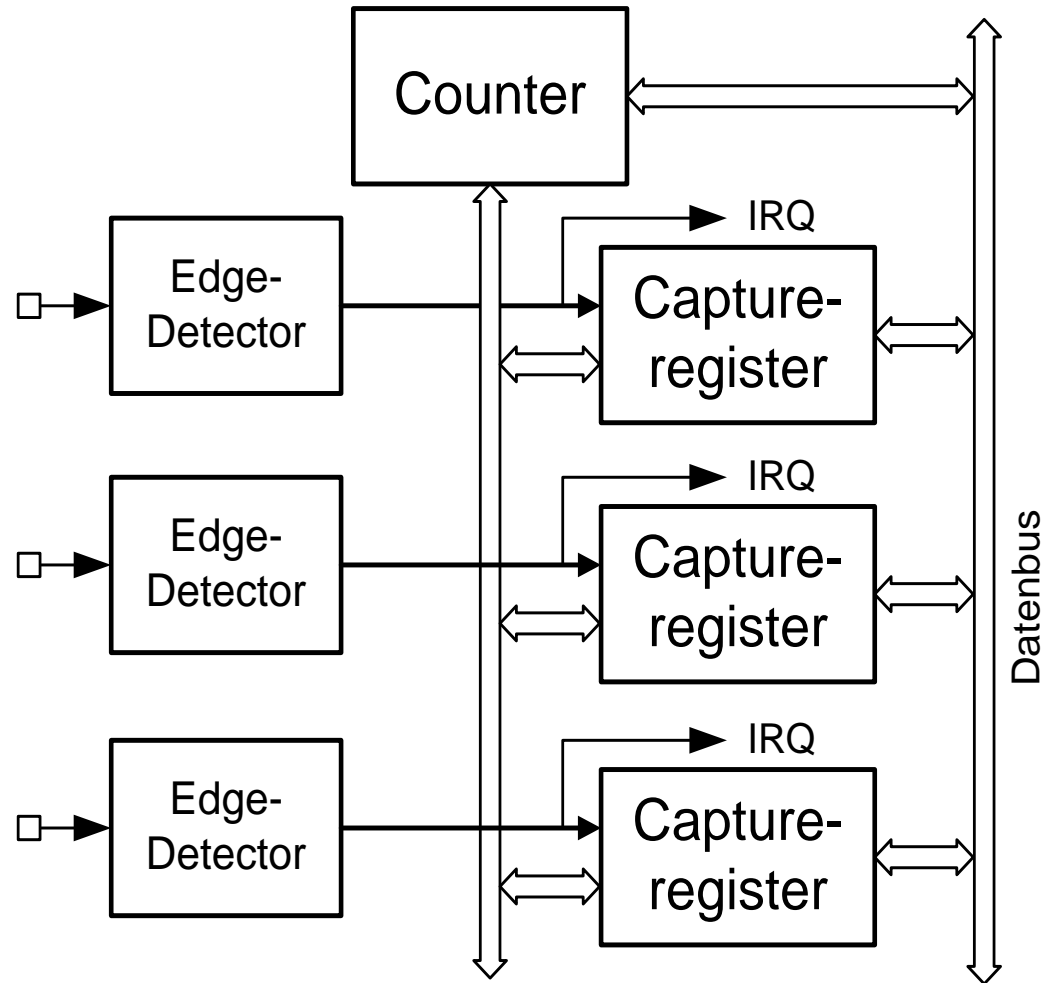
Zeitmessung



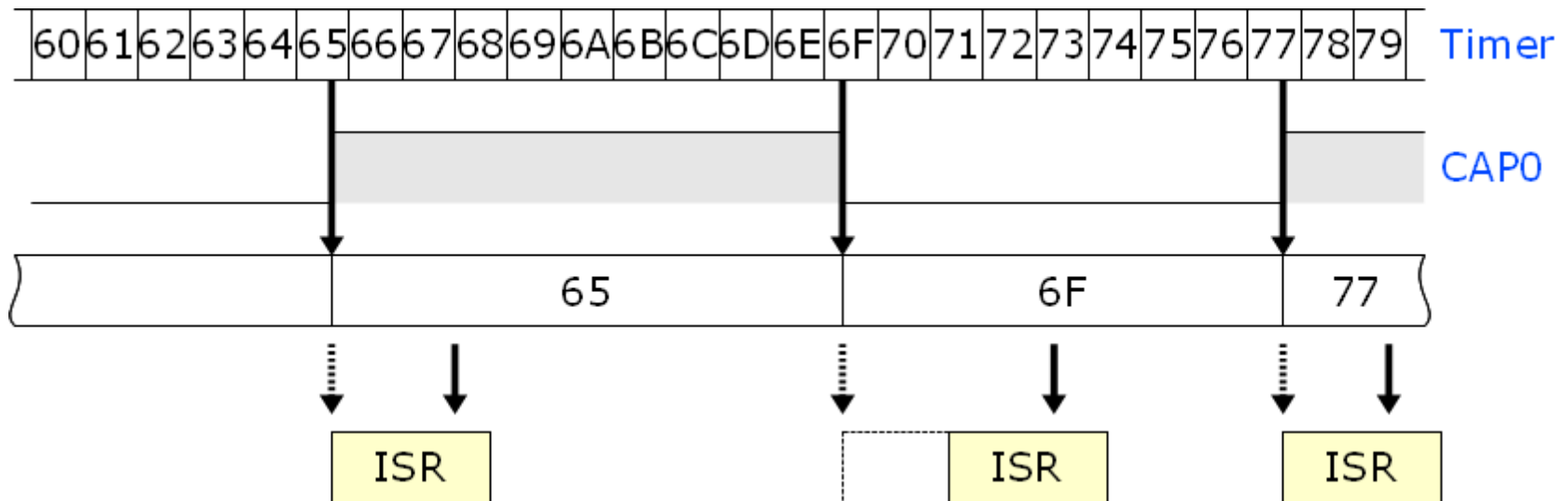
- ▶ **Idee:**
 - ▶ Externes Signal löst Interrupt aus.
 - ▶ In der ISR werden die Zeitpunkte mit Hilfe des intern getakteten Timers bestimmt.
- ▶ Aber Latenz- und Bearbeitungszeiten des Interrupts bewirken:
 - ▶ Ungenaue Messung
 - ▶ Jitter in Messergebnissen

Timer mit Input-Capture-Funktion

- Ermöglicht das taktgenaue Messen von externen Ereignissen.



Input-Capture-Funktion: Zeitmessung

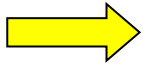


- ▶ Latenz- und Bearbeitungszeit des Interrupts haben keinen Einfluss auf die Messung:
 - ▶ taktgenaue Bestimmung möglich.
- ▶ Aber Ergebnis muss bis zum Eintreffen des nächsten Ereignisses gelesen werden.



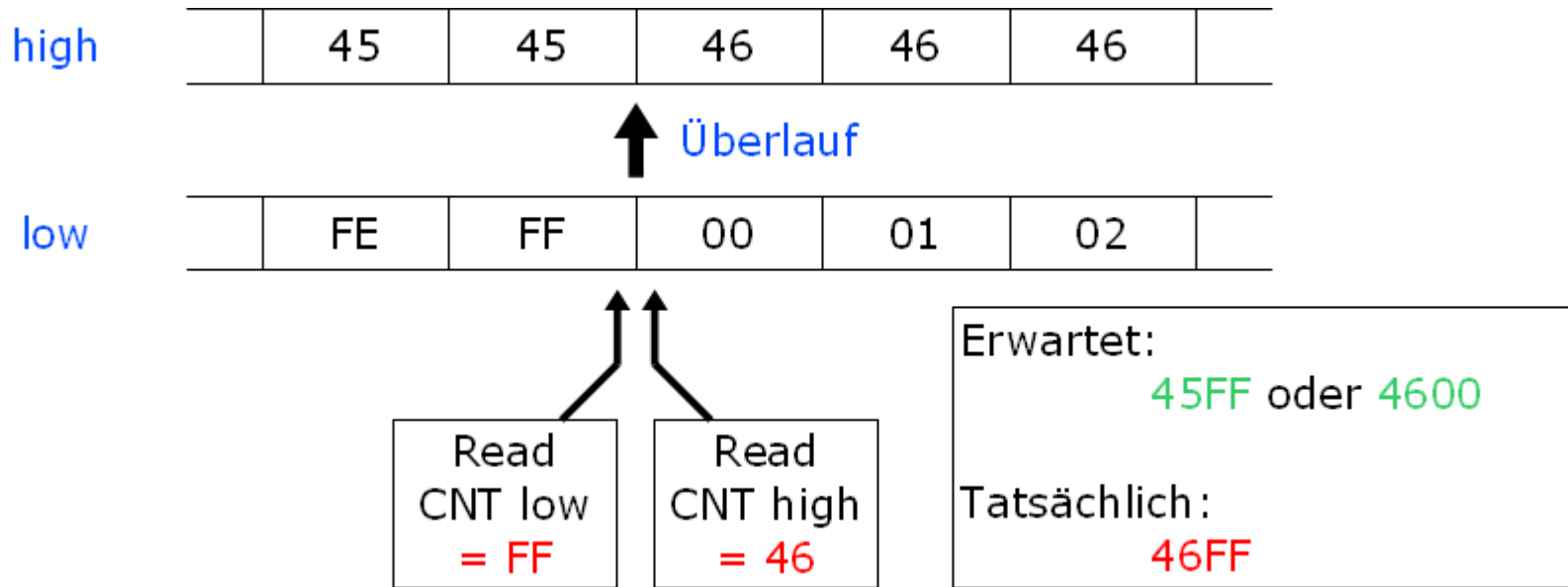
Übersicht

- ▶ Einleitung
- ▶ Output-Compare Funktion
- ▶ Input-Capture Funktion.
- ▶ Mehrfachzugriff auf Counter-Register.
- ▶ Timer/Counter im LPC2468





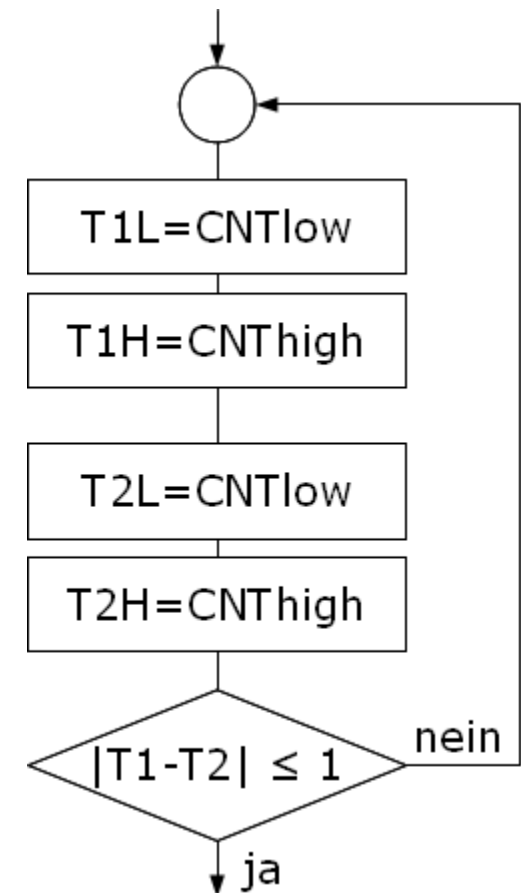
Zugriff auf 16-Bit Zähler mittels 8-Bit Bus



- ▶ Lesen des 16-Bit Registers erfordert zwei Zugriffe.
- ▶ Normalerweise Ungenauigkeit von 1
 - ▶ Bei gleichzeitiger Änderung des High-Registers:
 - Abweichung von 255

Zugriff auf 16-Bit Zähler mittels 8-Bit Bus: Sicherer Lesezugriff per Software

- ▶ **Problematisch:**
 - ▶ mindestens 4 Leseoperationen auf Register
 - ▶ Zeitliche Ungenauigkeit des Zugriffs
 - ▶ **Achtung:**
 - Unterbrechung möglich zwischen Lesezugriffen auf low und high durch IRQ!
- ▶ **Probleme bei Schreibzugriffe?**



LPC2468: Lese- und Schreibzugriffe unproblematisch,
da 32-Bit Zugriffe und 32-Bit Register (atomarer Zugriff möglich)



Übersicht

- ▶ Einleitung
- ▶ Output-Compare Funktion
- ▶ Input-Capture Funktion.
- ▶ Mehrfachzugriff auf Counter-Register.
- ▶ Timer/Counter im LPC2468



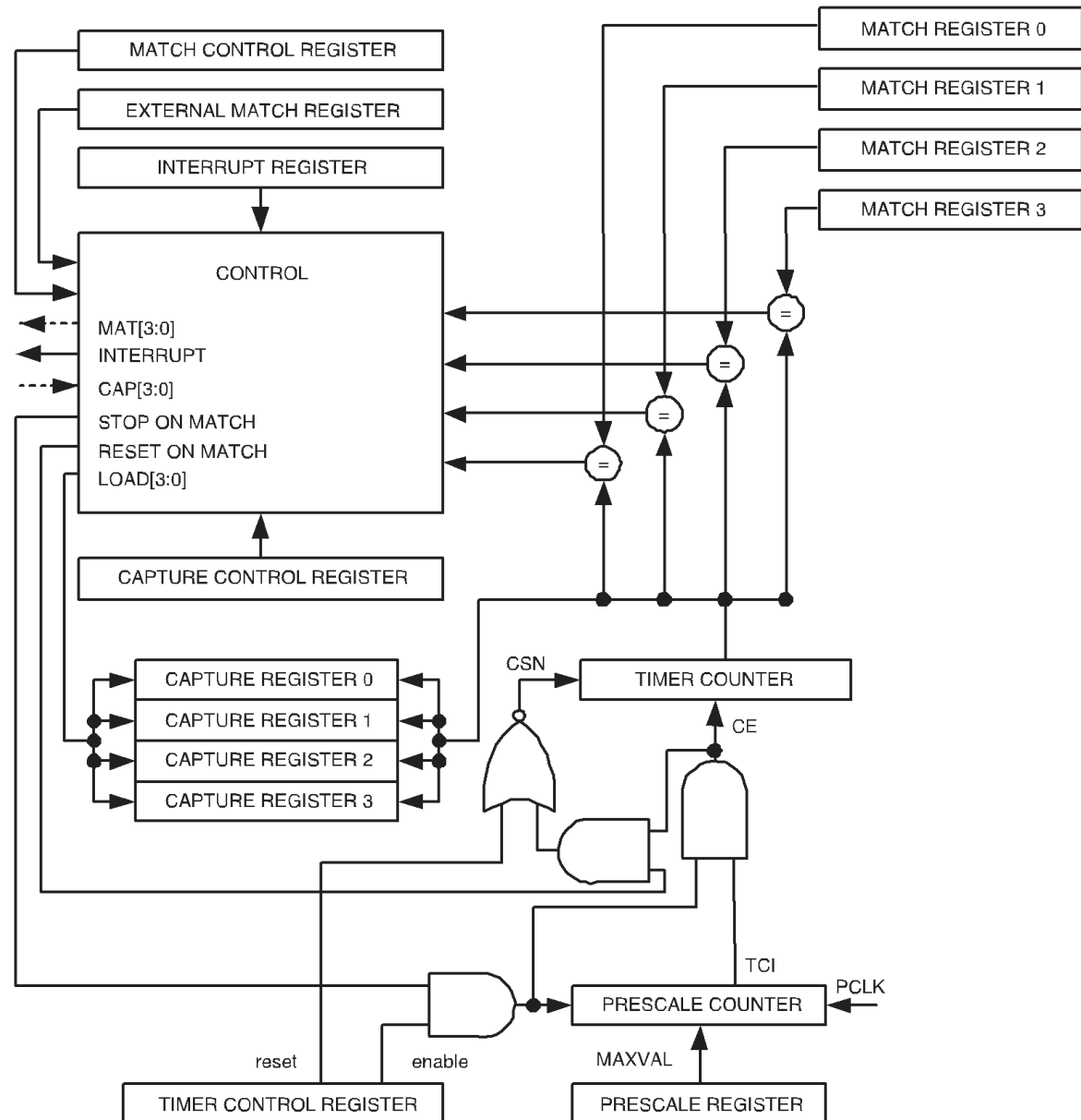


Timer-Bausteine im LPC2468

- ▶ **LPC2468 hat 4 Timer-Bausteine: Timer0 bis Timer3.**
- ▶ **Zusätzlich 2 „Pulse Width Modulators“ (PWM).**
- ▶ **Hohe zeitliche Auflösung:**
 - ▶ **Alle Register sind haben 32-Bit.**
 - ▶ **Maximale Taktfrequenz: Systemtakt (typisch 48 MHz).**
- ▶ **4 Capture-Register zum Erfassen externer Signale.**
- ▶ **4 Match-Register mit jeweils 3 Betriebsarten:**
 - ▶ **Kontinuierlicher Betrieb.**
 - ▶ **Stop beim Erreichen des Match-Wertes (Single Shot).**
 - ▶ **Zurücksetzen des Zählers beim Erreichen des Match-Wertes.**
- ▶ **4 Ausgänge / 4 Eingänge pro Timer möglich.**

CE WS12

Timer-Bausteine im LPC2468: Blockdiagramm





Timer-Bausteine im LPC2468: Datenregister

- ▶ **Timer Counter Register** **TIMERN_TC**
 - ▶ **Aktueller Zählerstand**
- ▶ **Prescale Register** **TIMERN_PR**
 - ▶ **Maximaler Zählerstand des Vorteilerzählers.**
- ▶ **Prescale Counter Register** **TIMERN_PC**
 - ▶ **Aktueller Zählerstand des Vorteilerzählers.**
- ▶ **Match Register** **TIMERN_MR[0/1/2/3]**
 - ▶ **Wert des Match Registers**
- ▶ **Capture Register** **TIMERN_CR[0/1/2/3]**
 - ▶ **Wert des Capture Registers.**



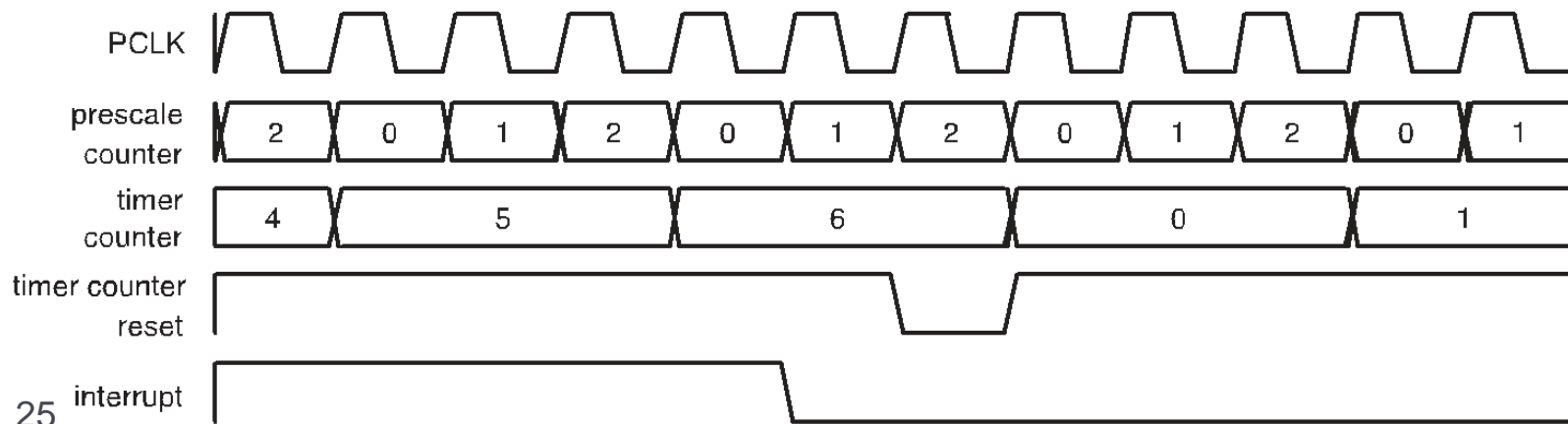
Timer-Bausteine im LPC2468: Steuerregister

- ▶ **Interrupt Register** **TIMERN_IR**
 - ▶ **Zeigt an, ob ein Match- oder Capture-Event aufgetreten ist.**
Muss per Software zurückgesetzt werden.
- ▶ **Timer Control Register** **TIMERN_TCR**
 - ▶ **Reset und Freigabe des Zählers.**
- ▶ **Count Control Register** **TIMERN_CTCR**
 - ▶ **Einstellung Betriebsmode: Zeitgeber oder Zähler.**
Auswahl des Zähleingangs.
- ▶ **Match Control Register** **TIMERN_MCR**
 - ▶ **Festlegung, was bei einem Match-Event passieren soll:**
Interrupt, Reset des Zählerstandes, Stop des Zählers.
- ▶ **Capture Control Register** **TIMERN_CCR**
 - ▶ **Festlegung, welche Flanke erfasst werden soll.**
Freigabe des Interrupts.
- ▶ **External Match Register** **TIMERN_EMR**
 - ▶ **Festlegung, was bei einem Match-Event mit dem externen Match-**
Ausgang (MATn.m) passieren soll: High, Low, Toggle

Timer-Bausteine im LPC2468: Anwendungsbeispiel 1

► Kontinuierlicher Zähler

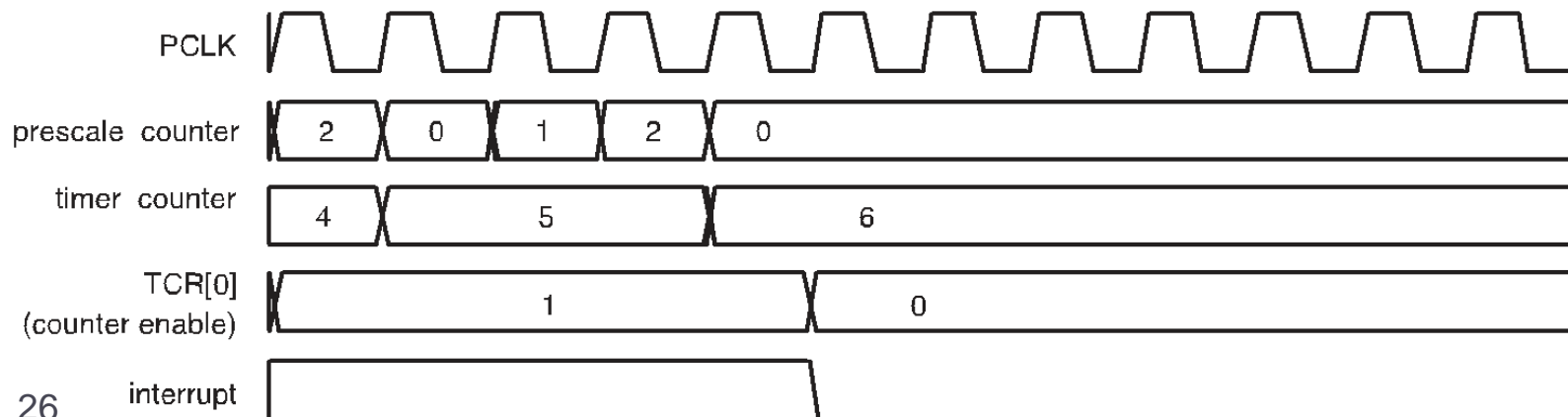
```
PCONP      |= (1<<22); // enable timer
PCLKSEL1   |= (1<<12); // set clock to cclk (48 MHz)
TIMER2_TCR = 0x02;     // reset timer
TIMER2_MR0 = 6;        // set match register 0
TIMER2_PR  = 2;        // set prescaler
TIMER2_IR   = 0xff;     // reset all interrupts
TIMER2_MCR  = 3<<0;     // interrupt on MR0, reset on MR0,
                        // do not stop timer on match
TIMER2_TCR  = 0x01;     // start timer
```



Timer-Bausteine im LPC2468: Anwendungsbeispiel 2

► Single Shot

```
PCONP      |= (1<<22); // enable timer
PCLKSEL1   |= (1<<12); // set clock to cclk (48 MHz)
TIMER2_TCR = 0x02;     // reset timer
TIMER2_MR0 = 6;        // set match register 0
TIMER2_PR  = 2;        // set prescaler
TIMER2_IR   = 0xff;     // reset all interrupts
TIMER2_MCR  = 5<<0;     // interrupt on MR0, no reset on MR0,
                        // stop timer on match
TIMER2_TCR  = 0x01;     // start timer
```



Timer-Bausteine im LPC2468: Anwendungsbeispiel 3

- ▶ LED2 auf dem TI-Board blinkt mit 1 Sekunde
- ▶ Verwendung des externen Match Signals MAT0.0

```
PCONP      |= (1<<1);    // enable timer
PCLKSEL0    |= (1<<2);    // set clock to cclk
PINSEL3     |= (3<<24);   // P1.28 is MAT0.0
TIMER0_TCR  = 0x02;       // reset timer
TIMER0_PR   = 0;          // set prescaler
TIMER0_IR    = 0xff;      // reset all interrupts
TIMER0_MCR   = 3<<0;      // interrupt on MR0, reset on MR0,
                          // do not stop timer on match
TIMER0_EMR   = 3<<4;      // toggle external match bit
TIMER0_TCR   = 0x01;      // start timer
TIMER0_MR0   = 48000000 /*Hz*/ / 2 /*Hz*/;
```