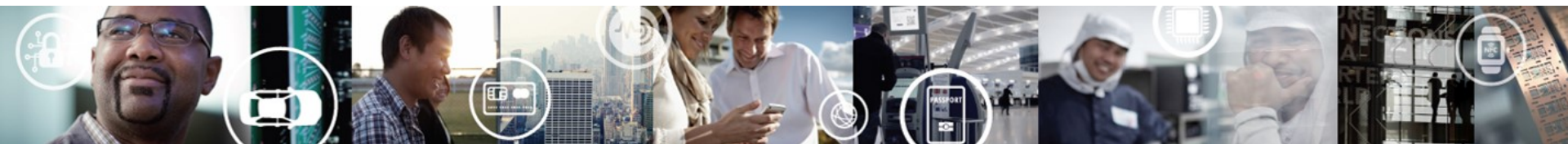


LPC82X 培训资料

状态可配置定时器

MAY, 2016



EXTERNAL USE



SECURE CONNECTIONS
FOR A SMARTER WORLD

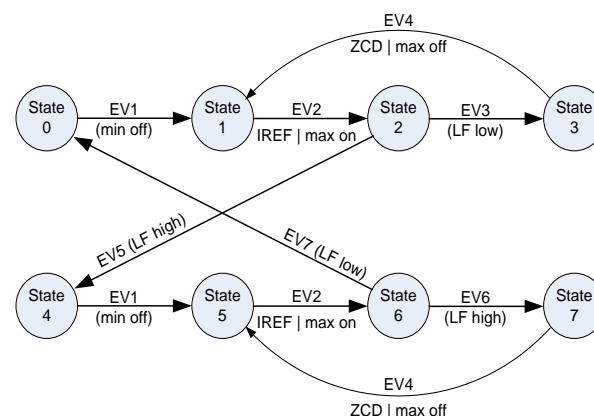
内容

- SCT概述
- SCT特性及时钟设置
- SCT输入/输出信号
- SCT事件分析
- SCT应用特性及实例分析

SCT概述

什么是SCT

- SCT：状态可配置计数器
- 可以作为普通Timer，硬件有限状态机引擎或两者功能的组合
- 普通Timer
 - 向上/向下计数
 - 计数值重载
 - 计数匹配
 - 中断产生
 - DMA触发
- 硬件有限状态机引擎
 - 可以灵活定义计数器，输出信号，中断，DMA行为
- 组合功能
 - Timer & I/O产生事件，事件控制Timer

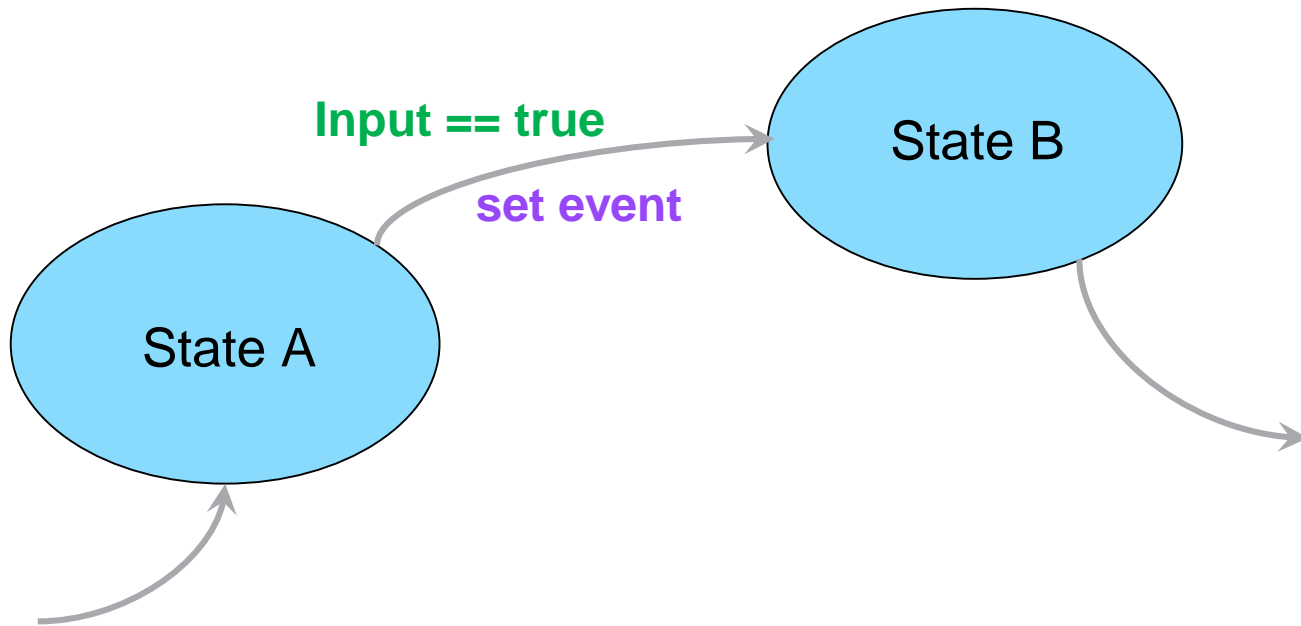


SCT基础块

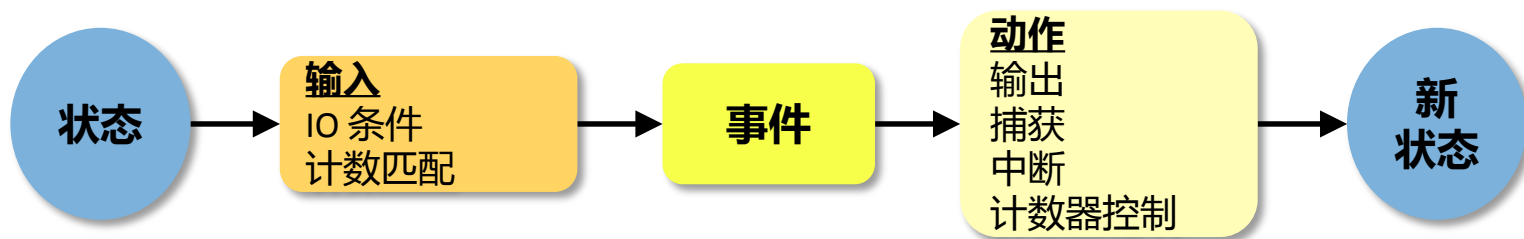
- Timer
 - 可以分为两个16位或者一个32位Timer
- 事件
 - 可以触发输出信号的转换，切换状态，改变计数器行为
- 状态
 - 定义了事件产生的背景条件
- 输入信号
 - 可以作为事件产生的一个条件
- 输出信号
 - 由事件触发产生，也可以作为事件产生的一个条件

什么是状态机

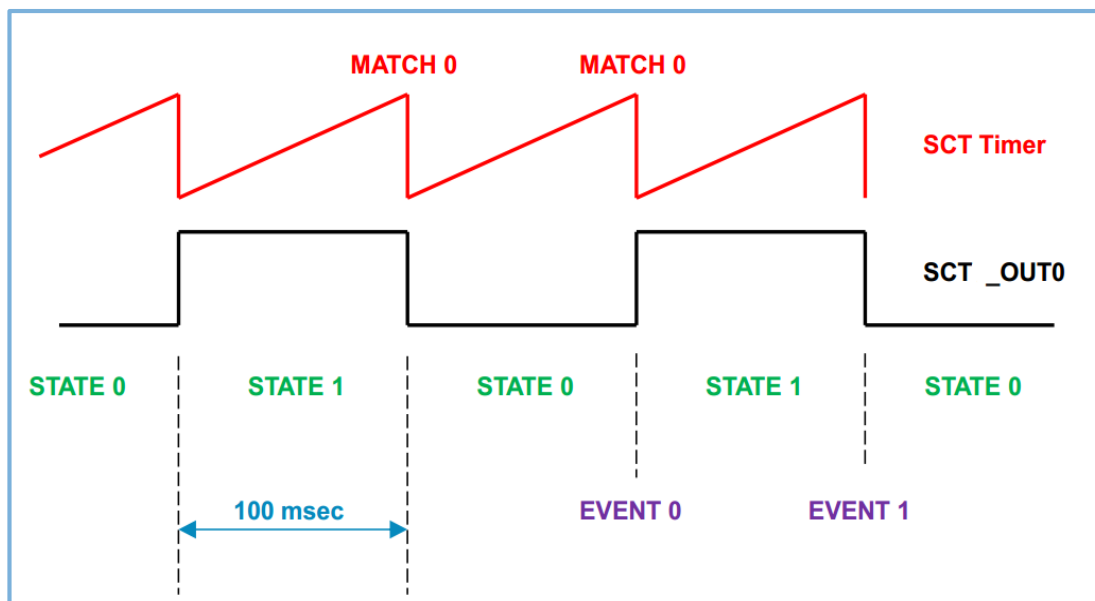
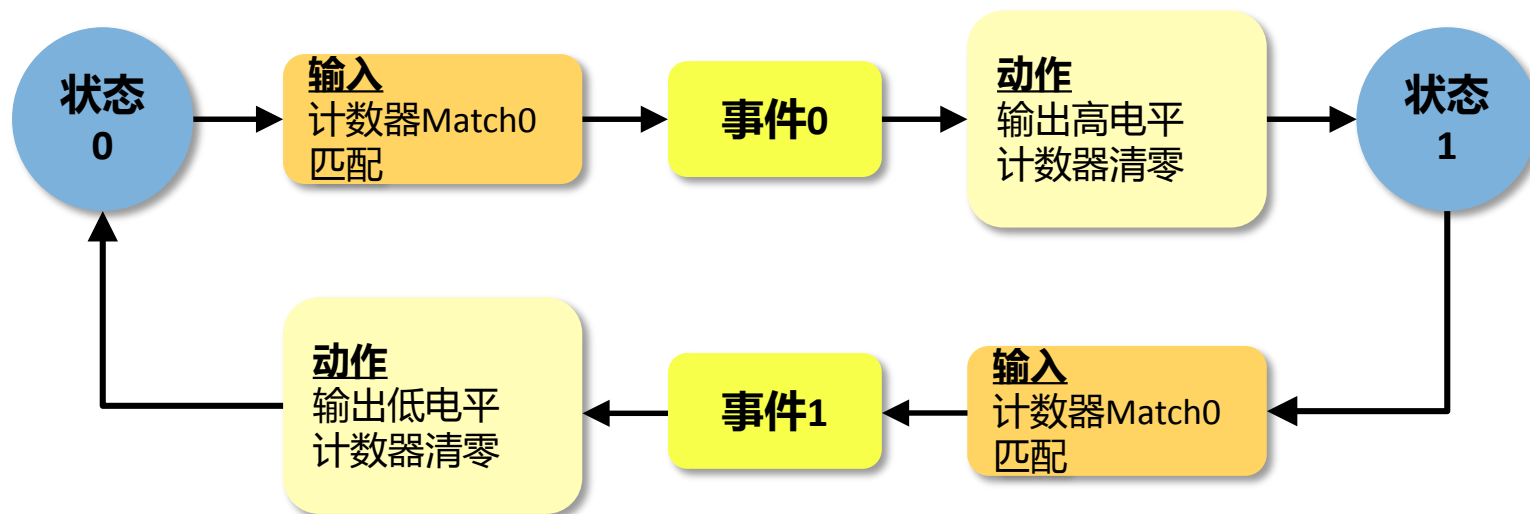
- 状态机组成：
 - 状态
 - 输入，比如输入/输出引脚信号或者计数器匹配
 - 输出（事件触发的动作）
 - 切换（状态变换）



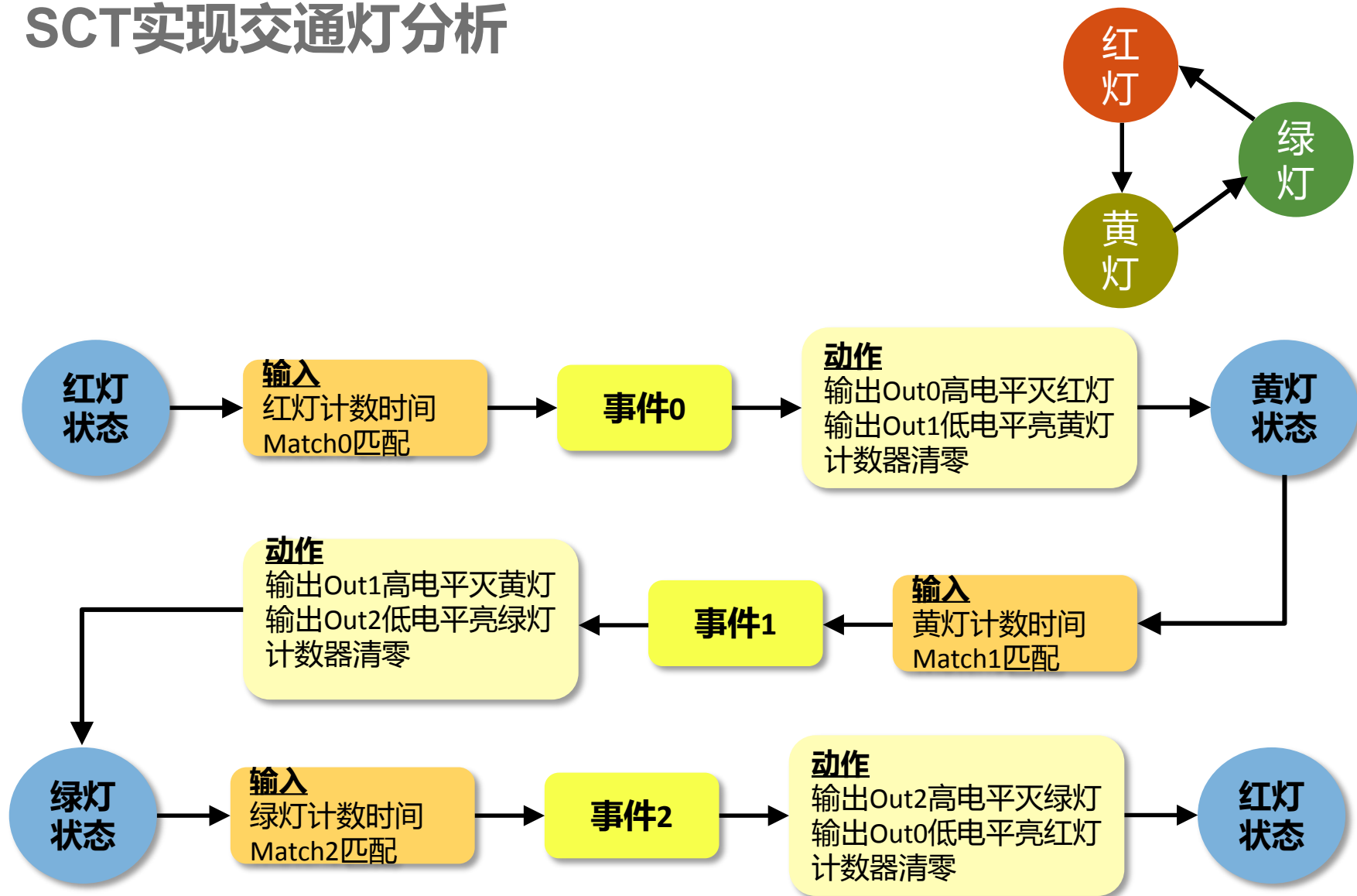
事件产生及状态切换顺序



SCT产生PWM分析

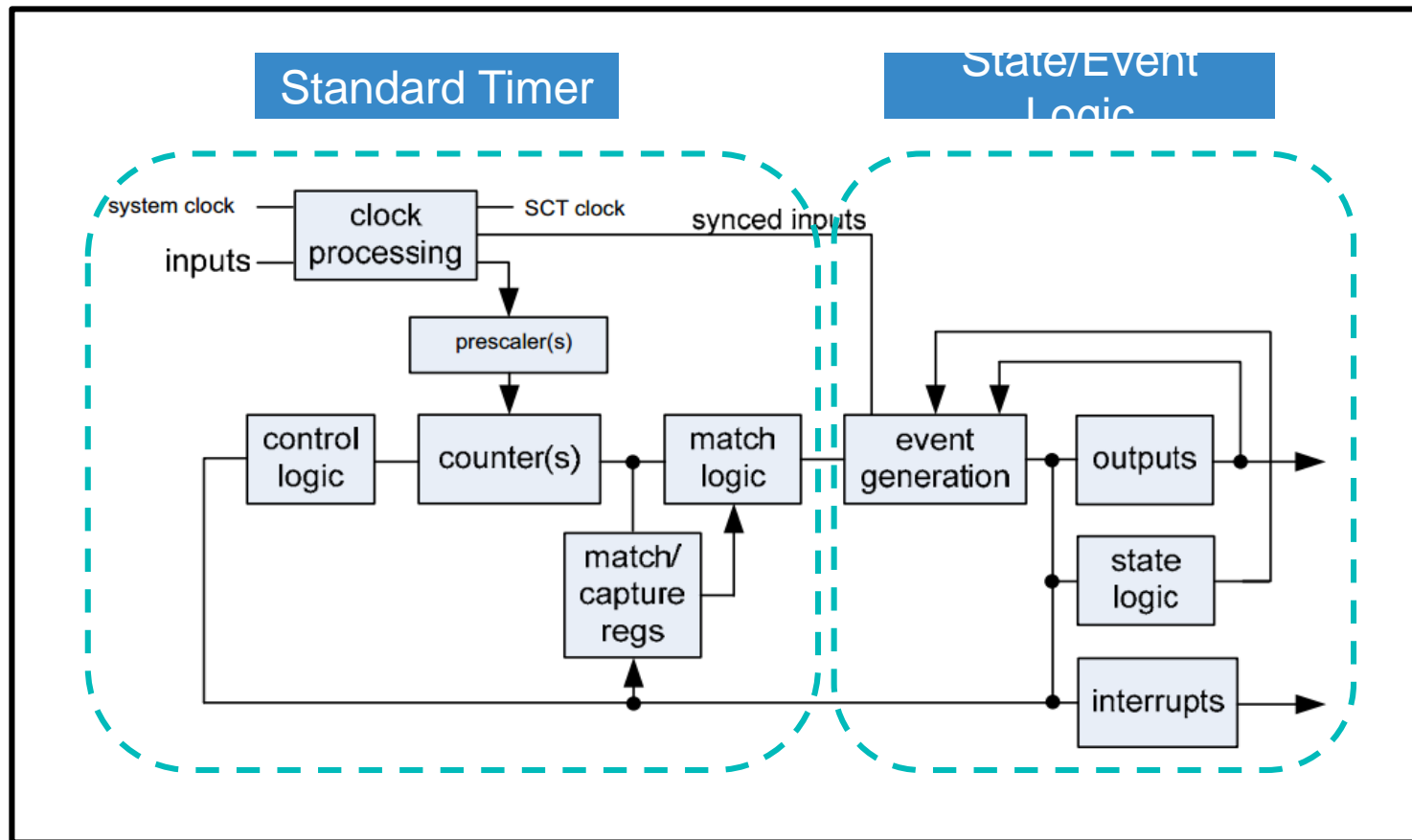


SCT实现交通灯分析



SCT功能框图

- 与传统的Timer比较，SCT增加了事件和状态(Event/State)的概念，对这两个概念深入分析后，就可以解开SCT的神秘面纱



状态机和SCT实现对应关系

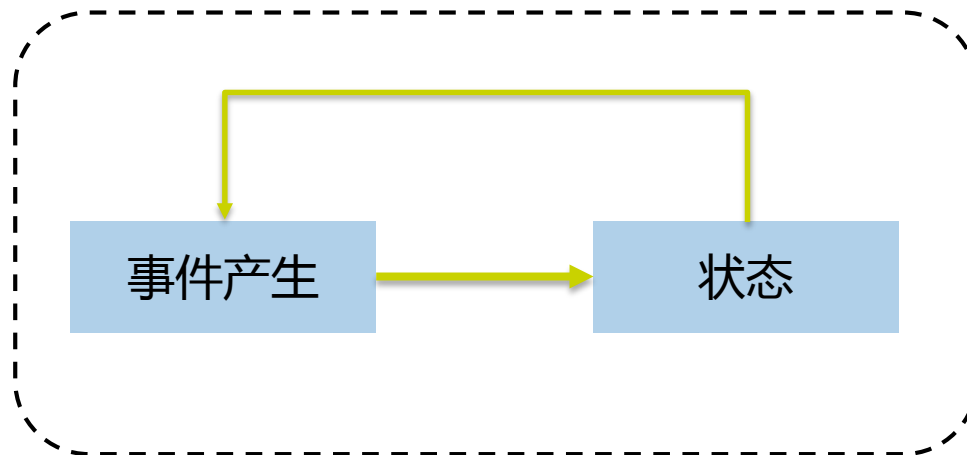
状态机元素	SCT实现
状态	<ul style="list-style-type: none">- 状态寄存器跟踪当前状态- 根据事件控制寄存器设置更新状态
输入	<ul style="list-style-type: none">-事件控制寄存器配置
输出	<ul style="list-style-type: none">- 由事件驱动，在输出置位寄存器和输出清零寄存器配置- 可作为状态切换的评估条件，在事件控制寄存器 设置
状态切换	<ul style="list-style-type: none">- 称为“事件”-事件控制寄存器定义-事件使能寄存器配置使能

SCT 事件

- 事件产生的输入条件或者输入源可以为：
 - 计数器匹配
 - 输入/输出信号：高低电平或者上升/下降沿
 - 计数匹配 [与|或] 输入/输出信号
- 任何事件可以：
 - 驱动输出信号
 - 切换状态机状态
 - 启动 / 停止 / 禁能 / 限制 计数器
 - 捕获当前计数器值
 - 产生中断或者DMA请求

SCT 状态

- 每个16位Timer都有特定的状态机
- 可以设定事件在某个具体的状态产生
- 状态可以明了地显示应用程序运行行为和SCT配置的关系
- SCT状态的概念，可以构造出无需软件干涉的硬件状态机，产生用户需求的复杂波形和定时任务



SCT 输入和输出信号

- 输入信号
 - 共有4个
 - 输入源可以是芯片内部信号或者外部引脚
 - 与SCT输入时钟同步
- 输出信号
 - 共有6个
 - 可以像输入信号一样作为事件产生的条件
 - 可以连接到片内其他模块，比如作为ADC触发信号，或者连接到片外引脚

SCT特性及时钟设置

LPC82x SCT特性

- SCT内部资源

Part	Inputs	Outputs	Match/Capture	Event	State
LPC82x	4	6	8	8	8

- Counter/Timer特性

- 可以配置为2个16-bit或者1个32-bit计数器
- Counter可以由系统时钟或者外部输入驱动
- 可以设置为单一向上计数或者up-down.
- Match和Capture寄存器共享，灵活选择使用Match或者Capture模式
- Match事件或者Input / Output toggle事件触发计数器捕获，并将计数值加载到Capture Register

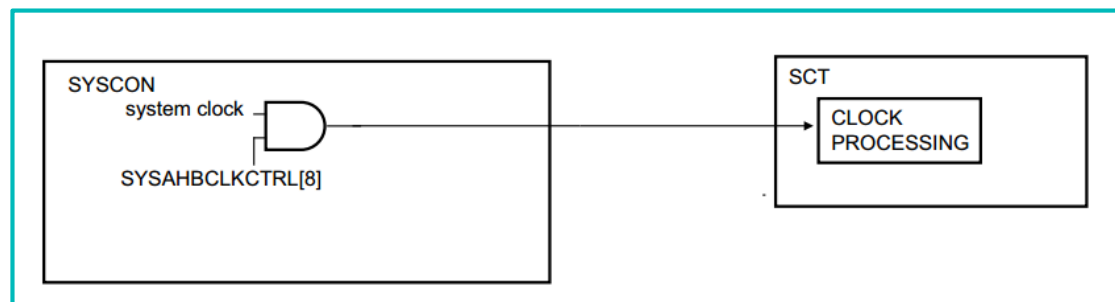
PWM特性

- 使用counter和Match寄存器来触发输出，产生周期和占空比可调的PWM信号
- 共可以输出6路占空比独立可调，统一周期的PWM信号

SCT 时钟设置

- SCT时钟使能

- 初始化SCT模块前，需要设置系统时钟控制寄存器使能SCT外设时钟



- 软件使能SCT外设时钟

- `Chip_Clock_EnablePeriphClock(SYSCTL_CLOCK_SCT);`

SCT 时钟模式设置

- 支持四种可选的时钟模式，包含一种完全的异步模式
- 可以选择任意一个SCT输入信号作为时钟源或者门控时钟
- 通过设置Configuration寄存器来选择时钟模式和SCT外部输入信号

-时钟模式设置

Configuration寄存器位		符号
2:1		CLKMODE
位值	时钟模式	描述
0x0	系统时钟模式	系统时钟驱动整个SCT模块，包括Counter和Counter分频器
0x1	采样系统时钟模式	系统时钟驱动SCT模块，Counter和Counter分频器由CKSEL位设定的Input信号边沿来驱动
0x2	SCT输入时钟模式	由系统时钟来同步，CKSEL位设定的Input信号边沿驱动SCT模块，包括Counter和Counter分频器
0x3	异步模式	CKSEL位设定的Input信号边沿驱动整个SCT模块，SCT输出信号也由Input时钟来同步，而不是系统时钟

SCT 外部输入时钟选择

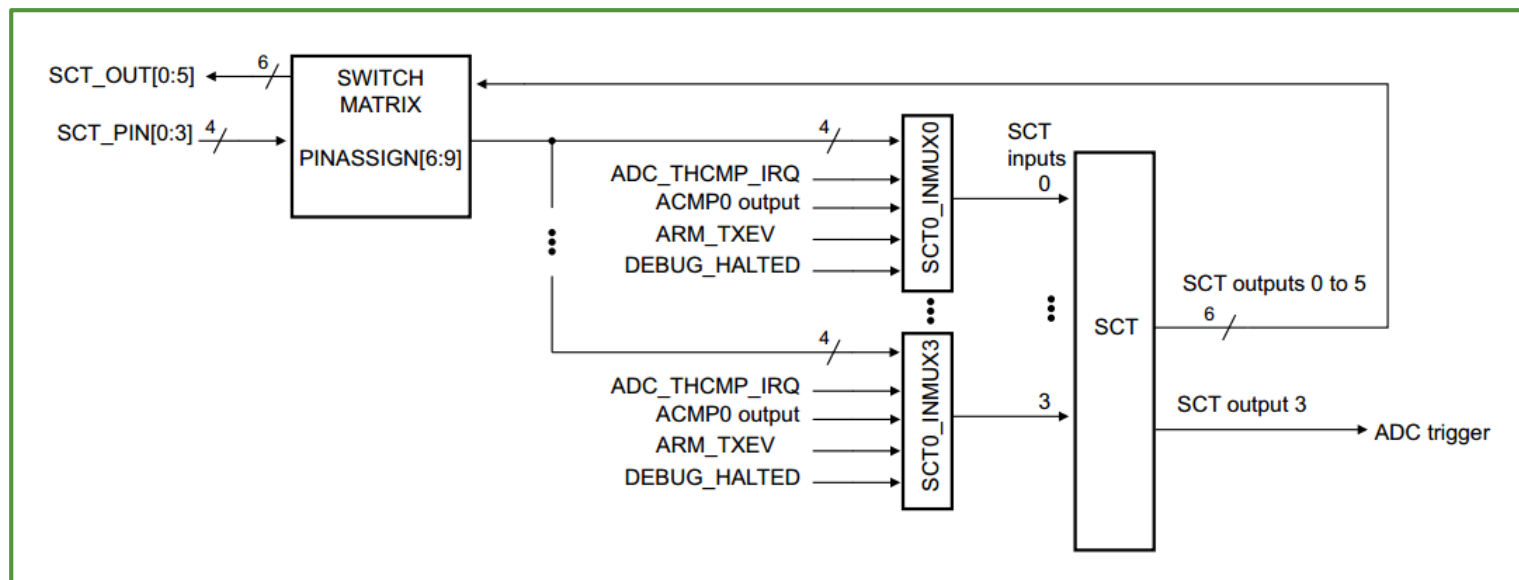
- 输入时钟选择

Configuration寄存器位		符号
6:3		CKSEL
位值	SCT时钟选择	描述
0x0	Input0上升沿	设定的Input信号及边沿作为SCT驱动时钟，首先要参考CLKMODE位的设定值
0x1	Input0下降沿	
0x2	Input1上升沿	
0x3	Input1下降沿	
0x4	Input2上升沿	
0x5	Input2下降沿	
0x6	Input3上升沿	
0x7	Input3下降沿	

SCT输入/输出信号

SCT 输入/输出信号框图

- SCT输入/输出为模块内部信号，可以连接片内其他模块信号，如SCT 输入信号连接模拟比较器输出或者ADC阈值比较输出，SCT输出信号可以作为ADC Trigger的输入信号，同时SCT输入/输出信号也可通过Switch Matrix连接芯片的外部引脚。
- SCT输入/输出信号连接图，SCT模块共有四个Input，六个Output



SCT 输入信号片内连接设置

- 首先设置Input Mux外设寄存器，分别选择四个输入信号连接片内其他模块信号或者片外引脚
 - //设置SCT_INPUT0信号连接到SCT_PIN0功能
 - Chip_INMUX_SetSCTInMux(LPC_INMUX, SCT_引脚NMUX_0, SCT_INP_IN0);

SCT0_INMUX[0:3]寄存器位		符号
3:0		INP_N
位值	连接信号	描述
0x0	SCT_PIN0	连接到SCT_PINx外部信号，需要通过Switch Matrix连接到外部Pin脚
0x1	SCT_PIN1	
0x2	SCT_PIN2	
0x3	SCT_PIN3	
0x4	ADC_THCMP_IRQ	连接到ADC模块阈值比较中断输出信号
0x5	ACMP_0	连接到模拟比较器 (Analog Comparator) 模块输出信号
0x6	ARM_TXEV	ARM核TXEV事件信号
0x7	DEBUG_HALTED	DEBUG_HALTED信号

SCT 输入信号片外连接设置

- 如果将SCT输入信号连接到SCT_PINx信号，还需配置Switch Matrix引脚分配寄存器PINASSIGN[6:7]连接到片外引脚
 - //设置SCT_PIN0到P0_1脚
 - Chip_SWM_MovablePinAssign(SWM_SCT_IN0_I, 1);

PINASSIGN[6]寄存器位	符号
31:24	SCT_PIN0_I
位值	描述
0x0-0x1C	分配SCT_PIN0功能信号到引脚，设定位值即为分配的引脚号，PIO0_0 (= 0) 到 PIO0_28 (= 0x1C)
PINASSIGN[7]寄存器位	符号
7:0	SCT_PIN1_I
15:8	SCT_PIN2_I
23:16	SCT_PIN3_I
位值	描述
0x0-0x1C	分配SCT_PIN[1:3]功能信号到引脚，设定位值即为分配的引脚号，PIO0_0 (= 0) 到 PIO0_28 (= 0x1C)

SCT 输出信号片外连接设置

- 需配置Switch Matrix引脚分配寄存器PINASSIGN[7:9]，连接到片外引脚

-//设置SCT_OUT0到P0_1脚

-Chip_SWM_MovablePinAssign(SWM_SCT_OUT0_O, 1);

PINASSIGN[7]寄存器位	符号
31:24	SCT_OUT0_O
PINASSIGN[8]寄存器位	符号
7:0	SCT_OUT1_O
15:8	SCT_OUT2_O
23:16	SCT_OUT3_O
31:24	SCT_OUT4_O
PINASSIGN[9]寄存器位	符号
7:0	SCT_OUT5_O
位值	描述
0x0-0x1C	分配SCT_OUT[0:5]功能信号到引脚，设定位值即为分配的引脚号，PIO0_0 (= 0) 到 PIO0_28 (= 0x1C)

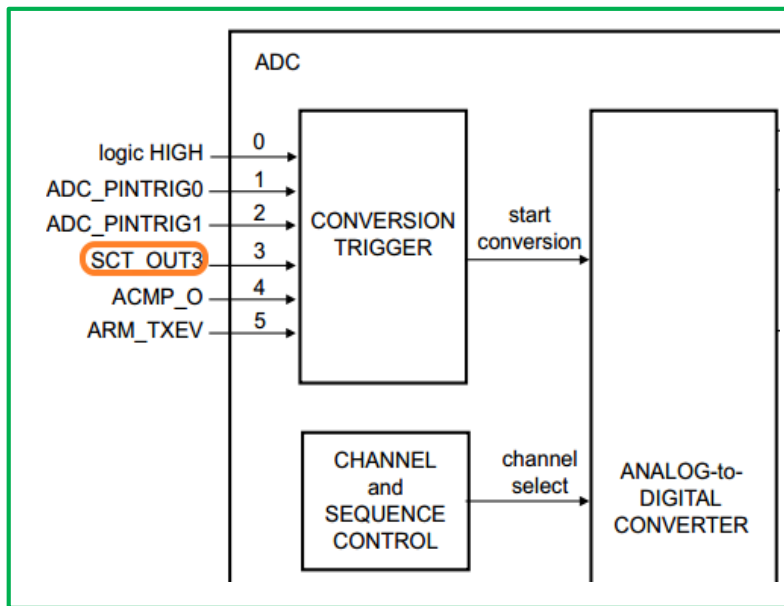
SCT 片外信号功能复用

- 通过Switch Matrix连接到外部引脚的SCT输入/输出信号，还需注意该引脚是否有复用功能，并且确保复用的功能信号处于禁能状态
- 设置引脚使能寄存器PINENABLE0, 来禁能复用功能
 - Chip_SWM_DisableFixedPin(SWM_FIXED_ACMP_I1);

位	符号	位值	描述
0	ACMP_I1	0	在PIO0_0引脚使能ACMP_I1功能
		1	禁能ACMP_I1功能
1	ACMP_I2	0	在PIO0_1引脚使能ACMP_I2功能
		1	禁能ACMP_I2功能
2	ACMP_I3	0	在PIO0_14引脚使能ACMP_I3功能
		1	禁能ACMP_I3功能
3	ACMP_I4	0	在PIO0_23引脚使能ACMP_I4功能
		1	禁能ACMP_I4功能
4	SWCLK	0	在PIO0_3引脚使能SWCLK功能
		1	禁能SWCLK功能
5:31	其他复用功能，请参考芯片用户手册		

SCT 输出信号片内连接设置

- SCT_OUT3还可以连接到ADC的ADC_trigger信号，触发ADC采样序列
- 通过设置ADC转换序列寄存器SEQ[A:B]_CTRL，来选择触发源
-Chip_ADC_SetSequencerBits(LPC_ADC, ADC_SEQA_IDX, (3 << 12));



SEQ[A:B]_CTRL寄存器位		符号
14:12		TRIGGER
位值	硬件触发源	描述
0	Logic High	逻辑高
1	ADC_PINTRIG0	ADC外部引脚0
2	ADC_PINTRIG1	ADC外部引脚1
3	SCT_OUT3	SCT输出信号3
4	ACMP_O	模拟比较器输出信号
5	ARM_TXEV	ARM核TX事件

SCT事件分析

SCT事件触发

- 事件触发定义主要通过事件控制寄存器EV[0:7]_CTRL和事件使能寄存器EV[0:7]_STATE设置
- 八个事件分别对应八个控制寄存器和八个使能寄存器

SCT事件触发	触发条件定义		对应寄存器
	Match Logic	计数器及计数匹配	事件控制寄存器定义
	Synced Input	同步后的SCT输入信号	
	Outputs	SCT输出信号	
	State Logic	SCT状态	事件使能寄存器定义

计数器及计数匹配条件

- 该条件输入通过事件控制寄存器EV[0:7]_CTRL定义

SCT event control 寄存器位		符号
3:0		MATCHSEL
位值	事件触发定义	描述
0到7	匹配寄存器[0:7]选择	选择事件对应的匹配寄存器，当计数器的计数值达到该匹配寄存器的数值时，触发该事件
SCT event control 寄存器位		符号
22:21		DIRECTION
位值	事件触发定义 /计数方向选择	描述
0	计数方向无关	事件触发与计数方向无关
1	向上计数	该事件只有在计数器向上计数时才会触发
2	向下计数	该事件只有在计数器向下计数时才会触发

输入\输出信号条件 I/O选择

- 该条件由事件控制寄存器EV[0:7]_CTRL定义

SCT event control 寄存器位		符号
5		OUTSEL
位值	输入/输出信号选择	描述
0	选择输入信号	选择输入信号作为该事件触发的条件
1	选择输出信号	选择输出信号作为该事件触发的条件
SCT event control 寄存器位		符号
9:6		IOSEL
位值	输入/输出信号值选择	描述
0到3，或者0到5	信号值	根据OUTSEL位值，当为0时，从0到3具体选择某个输入信号作为事件的触发条件；当为1时，从0到5具体选择某个输出信号作为事件的触发条件

输入\输出信号条件 触发沿/电平

- 该条件由事件控制寄存器EV[0:7]_CTRL定义

SCT event control 寄存器位		符号
11:10		IOCOND
位值	I/O触发电平或边沿	描述
0x0	低电平触发	根据OUTSEL和IOSEL位选择的I/O信号，IOCOND定义该信号的触发方式
0x1	上升沿触发	
0x2	下降沿触发	
0x3	高电平触发	

I/O和计数匹配组合条件

- 该条件由事件控制寄存器EV[0:7]_CTRL定义

SCT event control 寄存器位		符号
13:12		COMBMODE
位值	匹配和I/O条件的组合方式	描述
0x0	或	匹配和I/O条件相或后触发事件
0x1	只有匹配	只用匹配条件触发事件
0x2	只有I/O信号条件	只用I/O条件触发事件
0x3	与	匹配和I/O条件相与后触发事件

SCT状态条件

- 状态条件由事件使能寄存器EV[0:7]_STATE定义

SCT event Enable寄存器位		符号
7:0		STATEMSKn
位值	描述	
一位对应一个状态	八位分别对应SCT的八个状态，位0对应状态0，位1对应状态1。当置1时，该事件将在对应的状态使能；置0时，该事件将在对应的状态禁能。如果所有位全部设置为0，则该事件不对应任何状态，将被永久禁能。	

SCT事件驱动

- SCT所有的动作都由事件来驱动，是事件的输出结果
- SCT事件驱动列表及对应的设置寄存器如下

SCT事件驱动	驱动控制		对应寄存器
	Outputs	SCT输出信号	SCT output set register[0:5]
			SCT output clear register[0:5]
	State Logic	SCT状态值更新	SCT event control register[0:7] SCT state register
	Interrupts	SCT中断	SCT event interrupt enable register
	Match/Capture	SCT匹配/捕获动作	SCT capture control registers[0:7]
	Counter Control Logic	计数器控制	SCT limit event select register
			SCT halt event select register
			SCT stop event select register
			SCT start event select register

SCT输出信号置位/清零

- SCT模块共有6个输出信号SCT_Out[0:5]，每个输出信号对应一组输出控制寄存器OUT[0:5]_SET和OUT[0:5]_CLR，分别设置输出信号的置位和清零

SCT output set register寄存器位		符号
7:0		SET
位值	描述	
一位对应一个事件	八位分别对应SCT的八个事件，位0对应事件0，位1对应事件1。当置1时，对应事件触发时，该输出信号输出高电平；置0无操作	
SCT output set register寄存器位		符号
7:0		CLR
位值	描述	
一位对应一个事件	八位分别对应SCT的八个事件，位0对应事件0，位1对应事件1。当置1时，对应事件触发时，该输出信号输出低电平；置0无操作	

SCT状态值更新

- 事件触发时可以产生对当前SCT状态的更新，进行状态变换
- SCT状态寄存器**STATE**记录当前的SCT状态值
- SCT事件控制寄存器**EV[0:7]_CTRL**设置事件触发时对状态值的更新方式

SCT event control 寄存器位		符号
14		STATELD
位值	状态值更新方式	描述
0x0	当前状态值与位STATEV值相加得到新的状态值	注：当多个事件同时触发时，只有编号最高的事件才能有效更改当前状态值
0x1	直接将位STATEV值加载到状态寄存器作为新的状态值	
SCT event control 寄存器位		符号
19:15		STATEV
位值	描述	
0到31	根据STATELD值，将该值相加或者直接加载到状态寄存器，结果作为新的状态值	

SCT中断产生

- 设置SCT事件中断使能寄存器**EVEN**，可以产生事件触发中断

SCT event interrupt enable 寄存器位		符号
7:0		IEN
位值	描述	
一位对应一个事件	八位分别对应SCT的八个事件，位0对应事件0，位1对应事件1。当置1时，对应的事件触发时将产生SCT中断；置0时，事件触发时不会产生中断。	

计数器捕获

- 事件触发时可以产生对SCT计数器的捕获功能
- 设置SCT捕获控制寄存器CAPCTRL[0:7]

SCT capture control 寄存器位		符号
7:0		CAPCON
位值	描述	
一位对应一个事件	八位分别对应SCT的八个事件，位0对应事件0，位1对应事件1。当置1时，对应的事件触发时将捕获SCT计数器；置0时，事件触发时不产生捕获动作。	

计数器控制

- 事件触发时可以控制计数器的计数方向、计数清零、启动、停止或者禁能

事件触发	计数器控制	对应寄存器
	计数器清零或者反向计数	限制寄存器
	启动计数器	启动寄存器
	停止计数器	停止寄存器
	禁能计数器	禁能寄存器

计数限制

- 事件触发后，计数器将被清零或者开始反向计数
- 设置SCT限制寄存器LIMIT

SCT limit event select 寄存器位		符号
7:0		LIMMSK
位值	描述	
一位对应一个事件	八位分别对应SCT的八个事件，位0对应事件0，位1对应事件1。当置1时，对应的事件触发时将清零计数器；如果计数器被设置为双向计数时，将改变计数方向。置0时，事件触发时不产生计数控制动作。	

启动计数器

- 设置SCT启动寄存器START

SCT limit event select 寄存器位		符号
7:0		STARTMSK
位值	描述	
一位对应一个事件	八位分别对应SCT的八个事件，位0对应事件0，位1对应事件1。当置1时，对应的事件触发时将启动计数器；置0时，事件触发时不产生计数控制动作。	

停止计数器

- 设置SCT停止寄存器STOP

SCT limit event select 寄存器位		符号
7:0		STOPMSK
位值	描述	
一位对应一个事件	八位分别对应SCT的八个事件，位0对应事件0，位1对应事件1。当置1时，对应的事件触发时将停止计数器，计数器保持原值，事件触发仍然处于使能状态；置0时，事件触发时不产生计数控制动作。	

禁能计数器

- 设置SCT禁能寄存器HALT

SCT limit event select 寄存器位		符号
7:0		HALTMSK
位值	描述	
一位对应一个事件	八位分别对应SCT的八个事件，位0对应事件0，位1对应事件1。当置1时，对应的事件触发时将禁能计数器，事件触发处于禁能状态；置0时，事件触发时不产生计数控制动作。	

SCT应用特性及实例分析

SCT应用特性

- SCT计数器可以当作1个32位或者2个16位计数器使用，通过配置寄存器SCT configuration register的第0位UNIFY选择
- 相应的计数控制寄存器，如启动事件选择寄存器、停止事件选择寄存器、限制事件选择寄存器和禁能事件选择寄存器，都可作为1个32位或者2个16位使用，控制相应的计数器
- 匹配和捕获功能寄存器共享，通过设置SCT匹配/捕获模式寄存器SCT match/capture mode register选择
- SCT的输出信号由触发事件控制，如果事件对某个输出信号的设置相冲突，即同时使能置位和清零操作，则通过SCT conflict resolution register来设置最终输出

SCT应用实例分析

• Blinky Match

```
void SCT_Init(void)
{
    LPC_SCT->CONFIG |= 0x1;                                     // unified timer

    LPC_SCT->MATCH[0].U      = SystemCoreClock/10;              // match 0 @ 100 msec
    LPC_SCT->MATCHREL[0].U = SystemCoreClock/10;

    LPC_SCT->EV[0].STATE = 0x00000001;                          // ev 0 happens in state 0
    LPC_SCT->EV[0].CTRL  = (0 << 0) |                          // related to match 0
                          (1 << 12) |                          // match condition only
                          (1 << 14) |                          // STATEV is new state
                          (1 << 15);                          // STATEV[15] = 1

    LPC_SCT->EV[1].STATE = 0x00000002;                          // ev 1 happens in state 1
    LPC_SCT->EV[1].CTRL  = (0 << 0) |                          // related to match 0
                          (1 << 12) |                          // match condition only
                          (1 << 14) |                          // STATEV is new state
                          (0 << 15);                          // STATEV[15] = 0

    LPC_SCT->OUT[0].SET = (1 << 0);                             // event 0 sets SCT_OUT_0
    LPC_SCT->OUT[0].CLR = (1 << 1);                             // event 1 clears SCT_OUT_0
    LPC_SCT->LIMIT_L = 0x0003;                                  // event 0 and 1 are limits

    LPC_SCT->CTRL_L &= ~(1 << 2);                             // unhalt the timer}
}
```

SCT应用实例分析

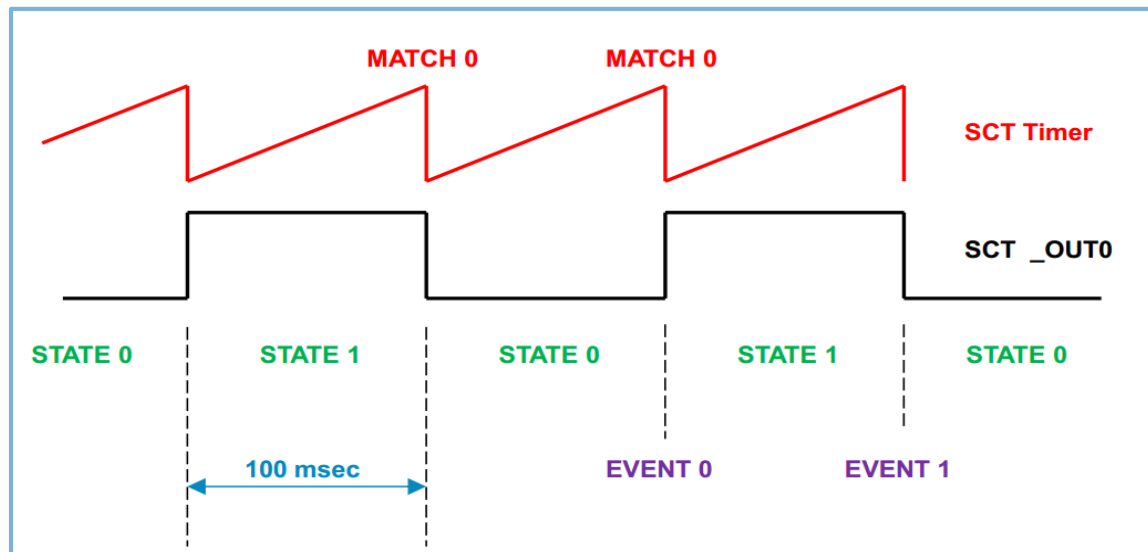
- Blinky Match配置

- 匹配寄存器：匹配寄存器0@100ms

- 输出信号：SCT_OUT0，连接到一个LED灯，控制100ms闪烁一次

- 事件：事件0和事件1，事件0控制SCT_OUT0输出高电平，事件1控制SCT_OUT0输出低电平，两个事件的输入条件都是Match0

- 状态：状态0和状态1，事件0在状态0下使能，事件1在状态1下使能



SCT应用实例分析

- Match Toggle

```
void SCT_Init(void)
{
    LPC_SCT->CONFIG |= (0x1 << 17) | 0x1;           // unified timer, auto limit

    LPC_SCT->MATCH[0].U    = SystemCoreClock/10;      // match 0 @ 100 msec
    LPC_SCT->MATCHREL[0].U = SystemCoreClock/10;

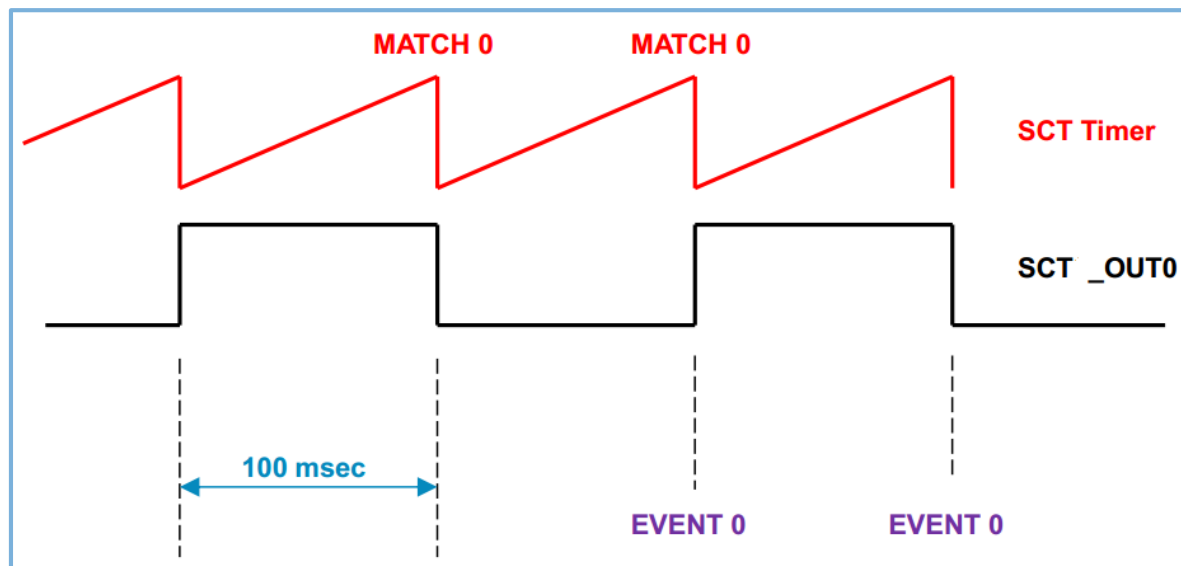
    LPC_SCT->EV[0].STATE = 0xFF;                      // ev 0 happens in all state
    LPC_SCT->EV[0].CTRL  = (0 << 0) |                // related to match 0
                          (1 << 12);                // match condition only
    LPC_SCT->OUT[0].SET  = (1 << 0);                  // event 0 sets SCT_OUT_0
    LPC_SCT->OUT[0].CLR  = (1 << 0);                  // event 0 clears SCT_OUT_0
    LPC_SCT->RES = 0x3;                               // output0 toggles on conflict

    LPC_SCT->CTRL_L &= ~(1 << 2);                    // unhalt the timer}
}
```

SCT应用实例分析

- Match Toggle配置

- 匹配寄存器：匹配寄存器0@100ms，自动清零计数器
- 输出信号：SCT_OUT0，事件0发生时变换输出电平，控制LED灯闪烁
- 事件：事件0，单一输入条件Match0，事件0触发时对SCT_OUT0控制冲突，切换输出电平
- 状态：没有使用





SECURE CONNECTIONS
FOR A SMARTER WORLD