

# Kinetis K10 在智能车中的应用

清华Freescale应用开发研究中心 于昊

2012年3月



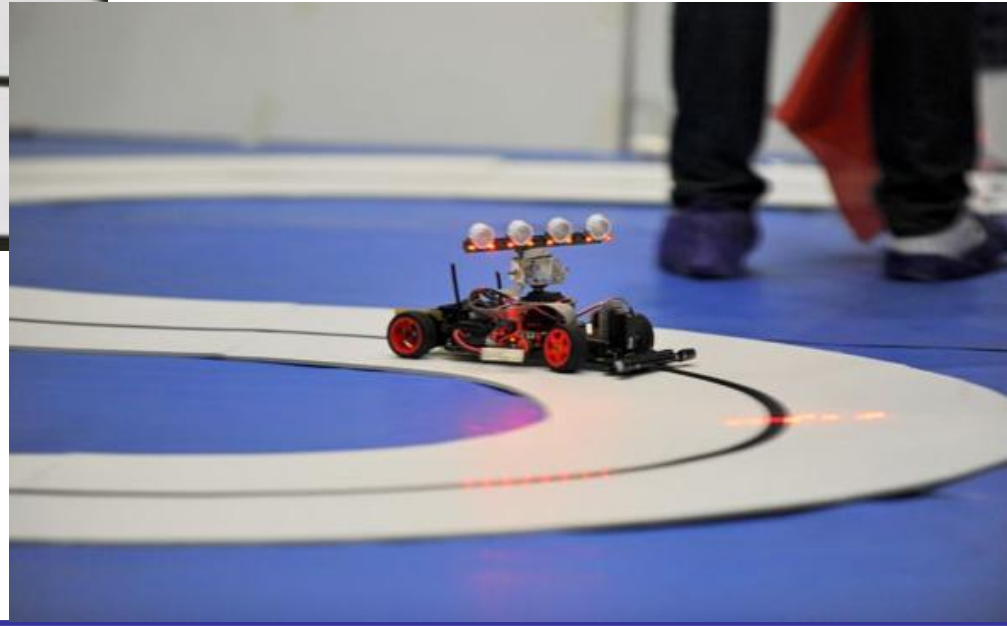
# Kinetis K10 在智能车中的应用

---

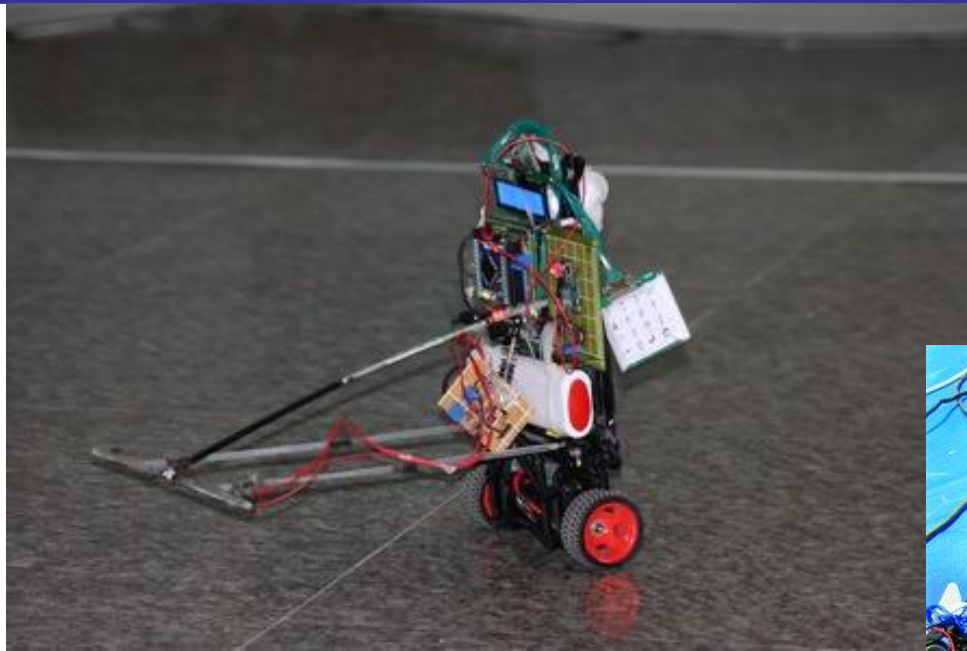
- 基本篇
  - 智能车的基本架构
  - 智能车的设计目标与需求
  - K10与XS128智能车相关特性比较
  - K10应用的注意事项
- 进阶篇
  - K10的DMA控制器在CMOS读出中的应用
  - K10的DSP功能的应用



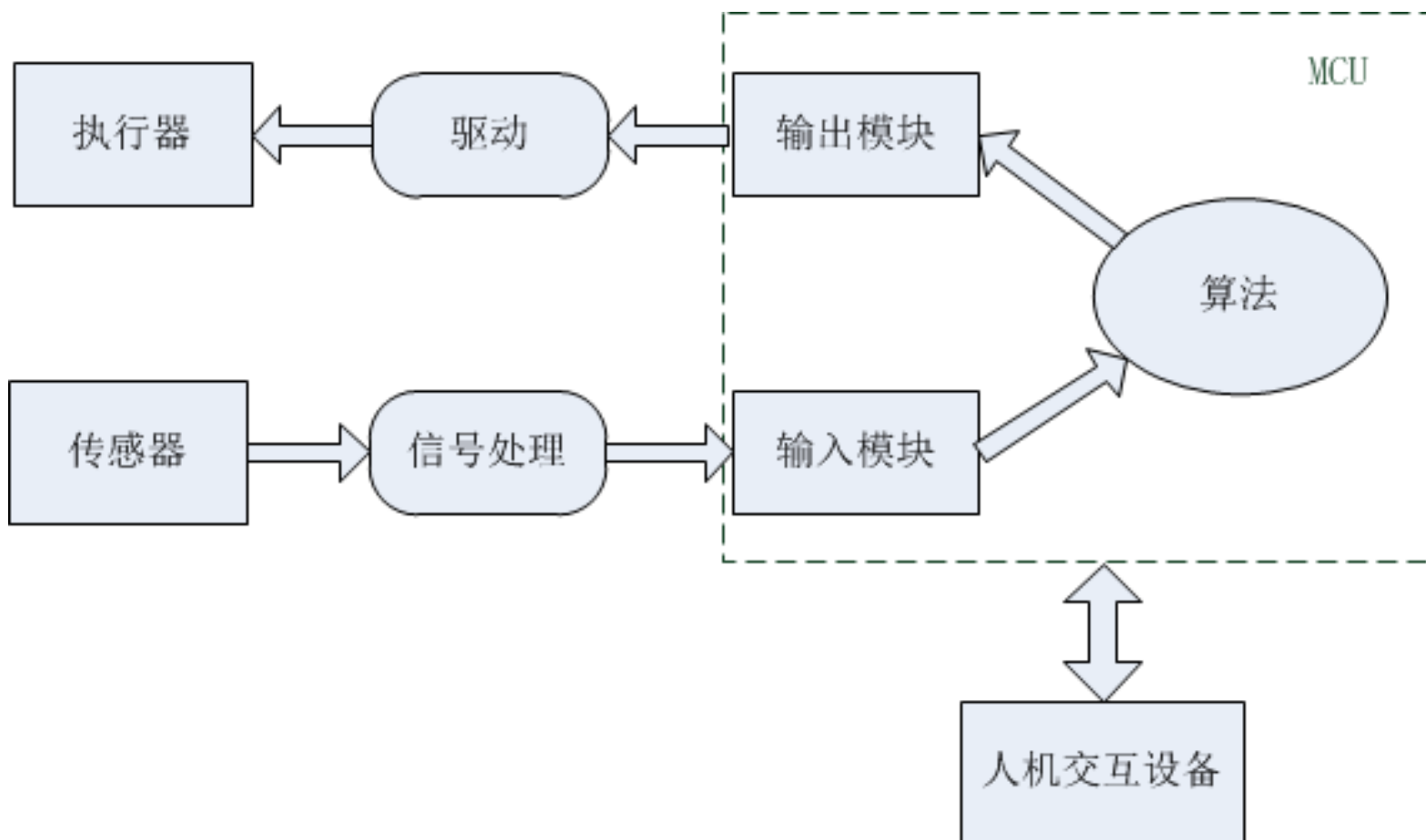
# 智能车的基本架构



# 智能车的基本架构



# 智能车的基本架构



# 智能车的基本架构

---

## ■ 传感器

摄像头(CCD,CMOS), 光电管, 电感线圈, 码盘, 加速度传感器, 陀螺仪.....

## ■ 执行器

舵机, 电机, 激光管, 底盘, 车轮.....

## ■ 人机交互设备

按键, 旋钮, 显示屏, SD卡, 无线模块, 串口, 调试器.....



# 智能车的设计目标

---

- 完成比赛用时最短
- 为了达到完成比赛用时最短的目标，传感器、执行器、人机交互界面、MCU有什么要求？



# 智能车的设计目标

---

## ■ 传感器

### ■ 看赛道更远

光电车的发展

### ■ 看赛道更精确

传感器的分辨率

### ■ 看赛道更详尽

摄像头、“点头”车

### ■ 看赛道更稳定

### ■ 姿态测得更准确

### ■ 测量速度更快





# 智能车的设计目标

---

- 执行器
  - 驱动电机能力更强
    - 电池、电机、驱动电路
  - 舵机响应更快与扭矩更大
    - 电压、PWM频率
  - 车轮抓地更好
    - 擦胎、磨胎
  - 整车更轻
  - 重心位置更合理
  - 底盘、齿轮、轴承等特性更好



# 智能车的设计目标

---

- 人机交互设备
  - 尽可能详尽的得到信息
  - 尽可能实时的得到信息
  - 尽可能快得传出信息
  - 尽可能方便的传入信息
- MCU
  - 优秀的算法
  - 与传感器、执行器、人机交互设备的通信灵活方便



# 对控制器的需求

---

- RAM足够大

  - 采集的数据多，控制算法可以更强大

- CPU处理能力够强

  - 在一个控制周期内可以处理更多的数据，用更复杂算法

- 最好有非易失存储器

  - 存储配置信息

- 读写外设模块尽可能快

- 读写外设模块尽可能不占用CPU时间

- 有一定的模拟量处理功能减小片外信号处理的要求



# K10与S12XS128比较

	K10(K10N512VLL100)	XS128(MC9S12XS128MAA)
封装	LQFP100(14mm*14mm)	QFP80(14mm*14mm)
可用IO	70	59
FLASH	512kbyte	128kbyte
RAM	128kbyte	8kbyte
非易失存储	无	8kbyte Dflash
CPU类型	ARM Cortex-M4	HCS12X
CPU最大速度	100M	40M
完整并行IO	PTC0-15 , PTD0-7	PTA0-7 , PTB0-7 , PTT0-7



# K10与S12XS128比较

	K10(K10N512VLL100)	XS128(MC9S12XS128MAA)
UART	6	2
SPI	3	1
I2C	2	0
Input Capture输入捕捉	Flex Timer模块，有硬件16bit计数器，至少8通道	通过中断计数，8通道
PWM	Flex Timer模块，至少8通道16bit	4通道16bit
SD卡控制器	有	无
外部中断	所有IO都可触发中断	部分IO可触发中断
DMA控制器	56个源，16通道	无



# K10与S12XS128比较

	K10(K10N512VLL100)	XS128(MC9S12XS128MAA)
ADC通道数	4组差分输入, 24个单端	8个单端输入
ADC精度	16bit	12bit
ADC最大速度	460ksps/16bit, 818ksps/13bit	415ksps/12bit
ADC特殊功能	可编程增益的放大器(PGA) 硬件平均功能(hardware average function)	无
DAC	12bit/2通道	无
模拟比较器	3通道	无
可编程基准电压输出	单通道1.2V/6bit	无



# K10与S12XS128比较

	K10(K10N512VLL100)	XS128(MC9S12XS128MAA)
开发环境	CW10.1	CW5.1
调试器	JTAG	BDM



# K10与S12XS128比较

---

- 用K10替代S12XS128，计算能力会有极大的提高
- RAM空间有了大幅的增加，可适应复杂算法
- 有更多的UART、SPI、IIC通信接口，可以挂更多外设，更便于调试
- PWM和脉冲计数功能比S12更强大、更方便好用
- 外部IO中断的使用更自由
- PCB板设计可以更集成，能省去SD卡驱动芯片和模拟信号的处理电路
- DMA功能可以让摄像头车的设计大为改变





# 用K10的注意事项

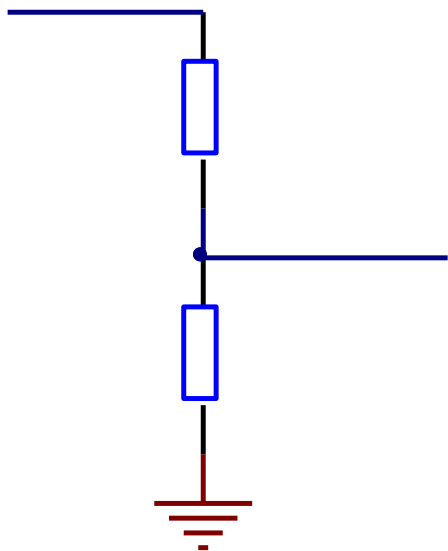
---

- K10供电是1.7V-3.6V，但很多外设是5V供电。
- K10的IO可以耐受5V电压，因此5V的外设输出高电平信号，K10是能正常接收不出问题的。
- K10在3.3V下低电平输入上阈是1.16V，即1.16V以下的信号会认为是0。
- 标准TTL输出低电平上限是0.4V，5V供电的CMOS输出低电平上限0.5V，即外设输出0的最高电压为0.5V。
- 5V外设输出低电平到K10时一般不会有问题，但在有噪声时相对更容易误触发。



# 用K10的注意事项

- 如果有问题可以通过电阻分压来简单解决。



# 用K10的注意事项

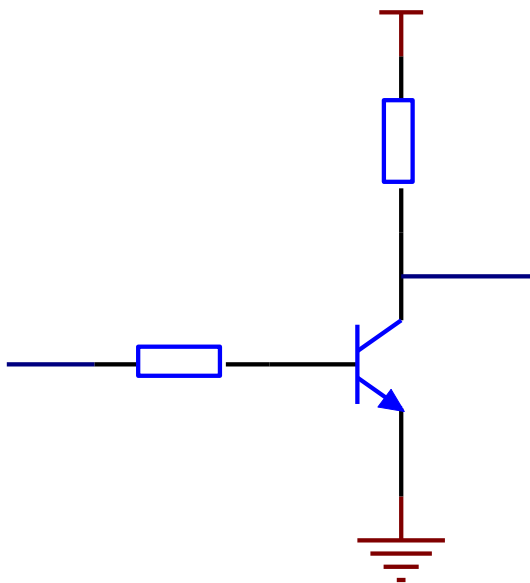
---

- K10向5V外设输出低电平信号时电压比5V外设所需的还要低，因此是没有问题的。
- 在3.3V下K10输出高电平下限是2.8V，即输出1时最小电压可能会是2.8V。
- 标准TTL输入高电平下阈是2V，即2V以上电压便认为是1。
- 5V供电的CMOS输入高电平下阈是3.5V，即3.5电压以上才认为是1。
- K10和TTL电平的外设通信会容易受到噪声干扰，与CMOS电平的外设就无法正常通信了，必须进行电平转换。



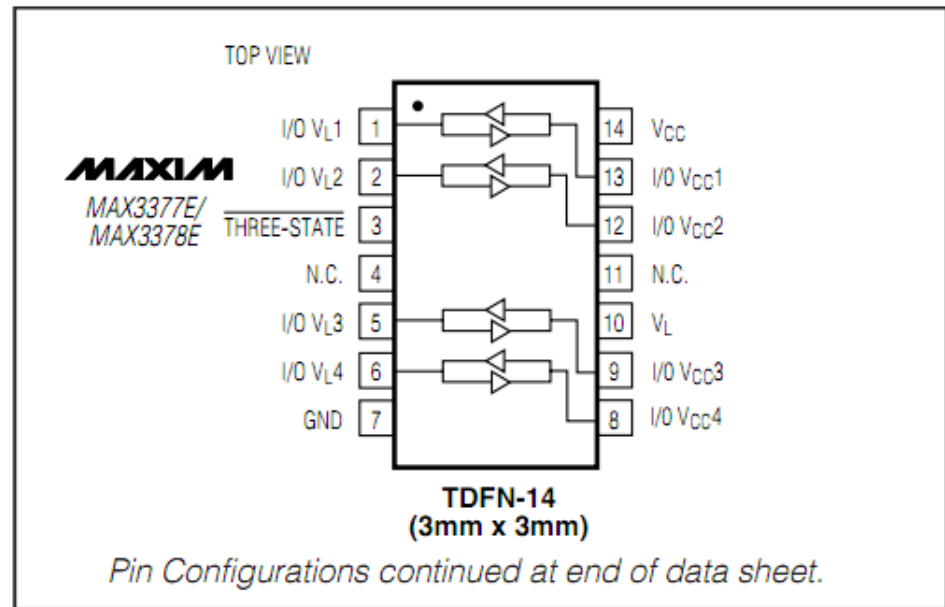
# 用K10的注意事项

- 可以用三极管进行简单的电平转换。



# 用K10的注意事项

- 在高速信号或者双向传输情况下要使用专用的电平转换芯片，如MAX3373，转RS232的话可用MAX3232。



19-2328; Rev 2; 11/07

**±15kV ESD-Protected, 1μA, 16Mbps, Dual/Quad  
Low-Voltage Level Translators in UCSP**



# DMA控制器在CMOS图像读出中的应用

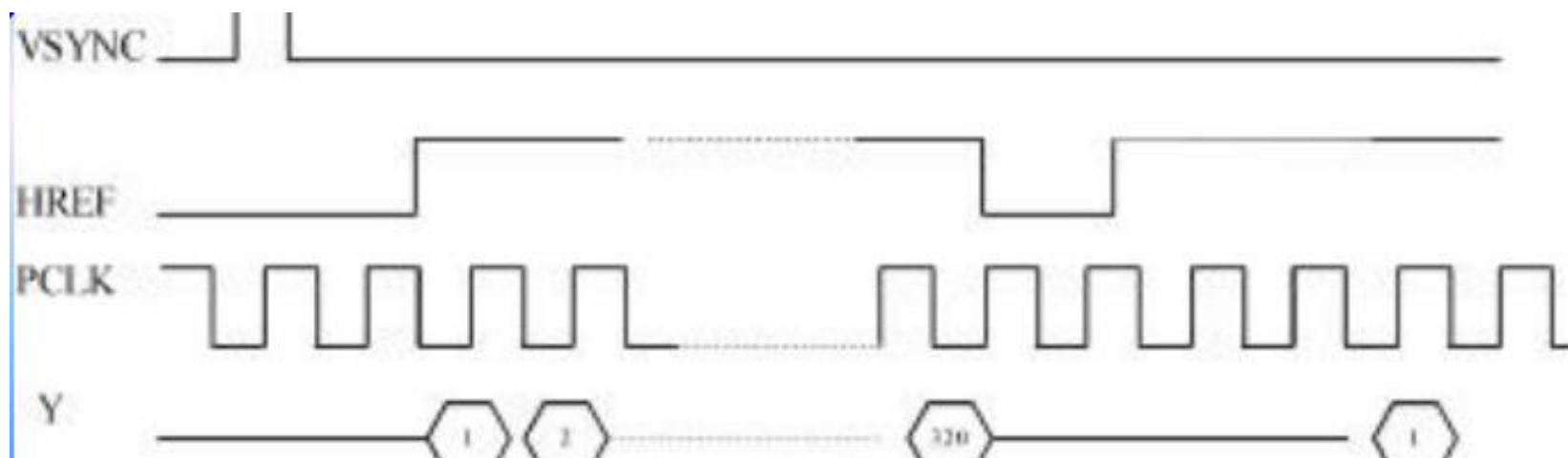
---

- 实验对象是基于ov7620芯片设计的数字摄像头hq7620，为1/3英寸CMOS摄像头，像素为640×480像素，输出8位图像信号，帧速率为60fps，并提供场同步、行同步以及像素时钟等多路同步信号。



# DMA控制器在CMOS图像读出中的应用

## ■ 信号波形



# DMA控制器在CMOS图像读出中的应用

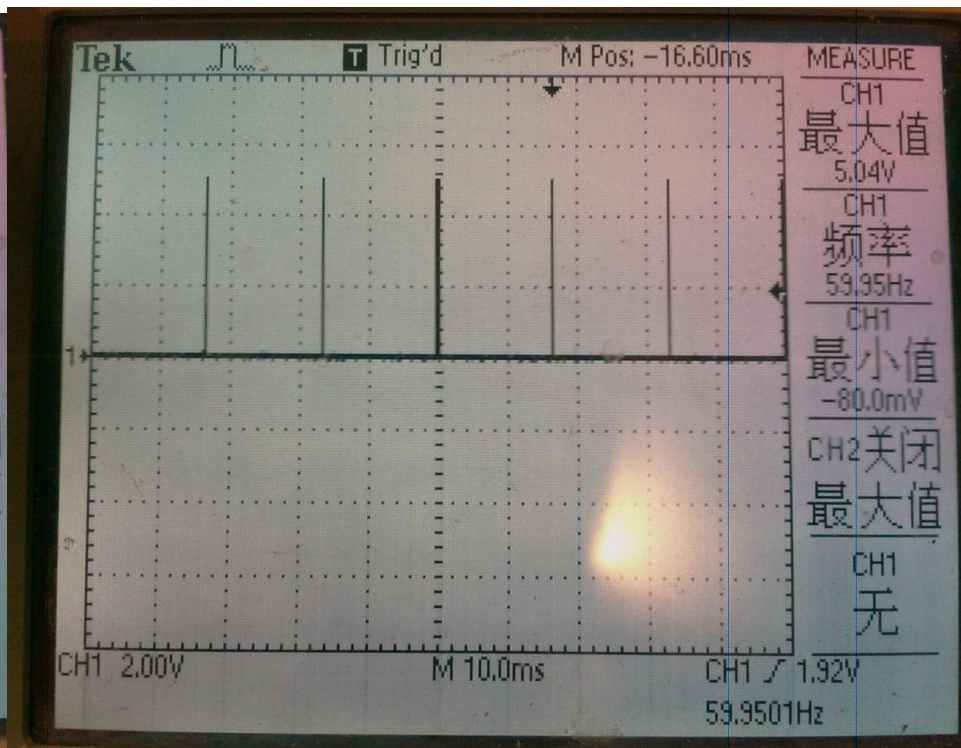
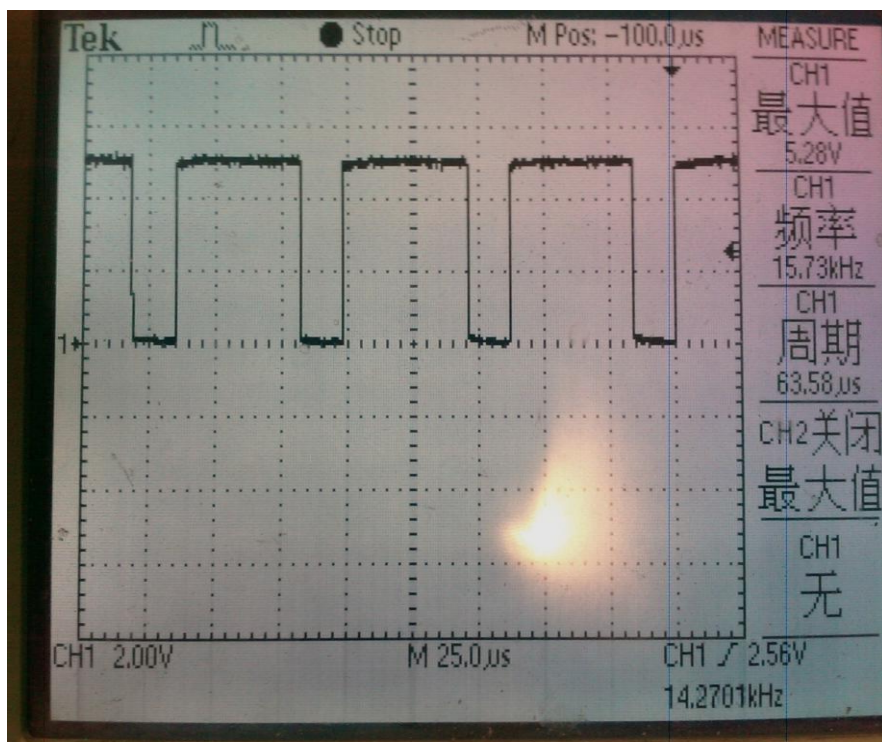
---

- 其中帧同步信号为60Hz，宽度。每一周期有240个行同步信号。
- 行同步信号正频宽47.4us，负频宽16.15us，每一周期有640个时钟信号。
- 时钟信号周期为74ns。
- 数据总线宽度8bit。





# DMA控制器在CMOS图像读出中的应用



# DMA控制器在CMOS图像读出中的应用

---

- 如何采集？
- 在不用外加硬件处理情况下，只能直接通过IO口读取。
- 在行同步信号触发中断后，不断的读取IO信息到内存。
- 受限于内存和读取IO速度，XS128单片机只能隔几行采集一行的数据，隔几帧采集一帧的数据。
- XS128在32M的CPU时钟下，用汇编以最优的方法来写读IO的程序，在一行能采到120个点左右。
- 若采集所有帧所有行，RAM空间不够大，CPU占用率会接近90%。
- K10来做图像采集有什么好办法?使用DMA控制器。



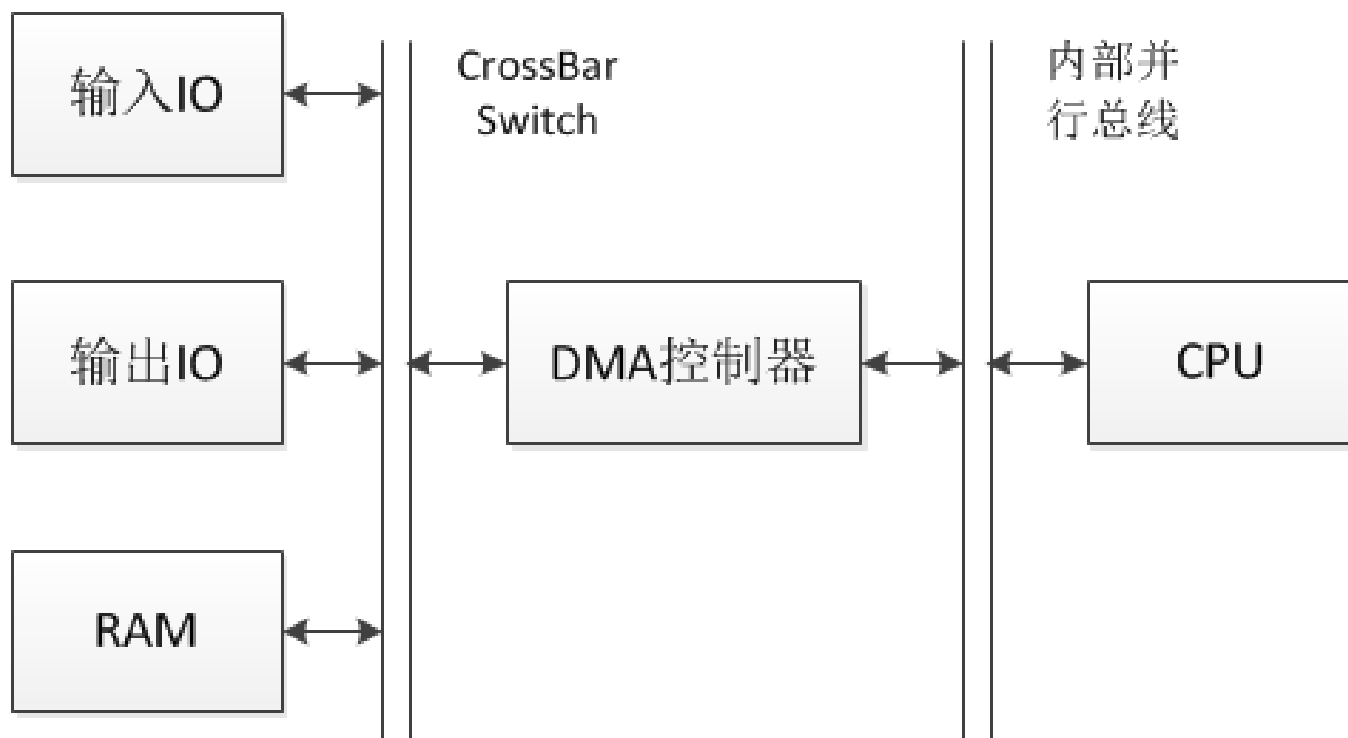
# DMA控制器在CMOS图像读出中的应用

---

- DMA(Direct Memory Access), 直接内存访问, 是一种不经过CPU而直接从内存存取数据的数据交换模式。在DMA模式下, CPU只须向DMA控制器下达指令, 让DMA控制器来处理数据的传送。



# DMA控制器在CMOS图像读出中的应用



# DMA控制器在CMOS图像读出中的应用

---

- K10的DMA功能主要由两个功能模块实现：  
DMAMUX(Direct memory access multiplexer)  
eDMA(The enhanced direct memory access controller)



# DMA控制器在CMOS图像读出中的应用

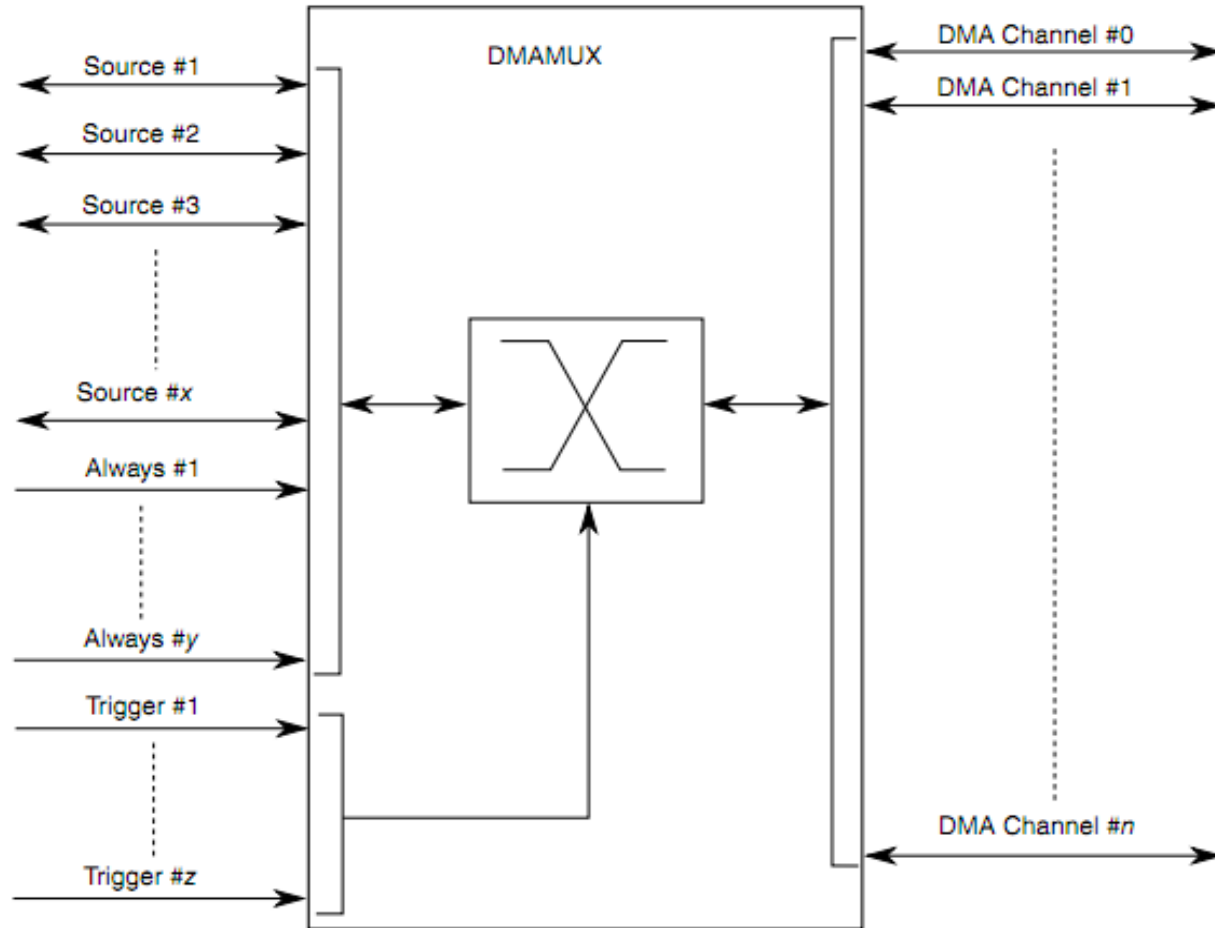


Figure 20-1. DMA MUX block diagram



# DMA控制器在CMOS图像读出中的应用

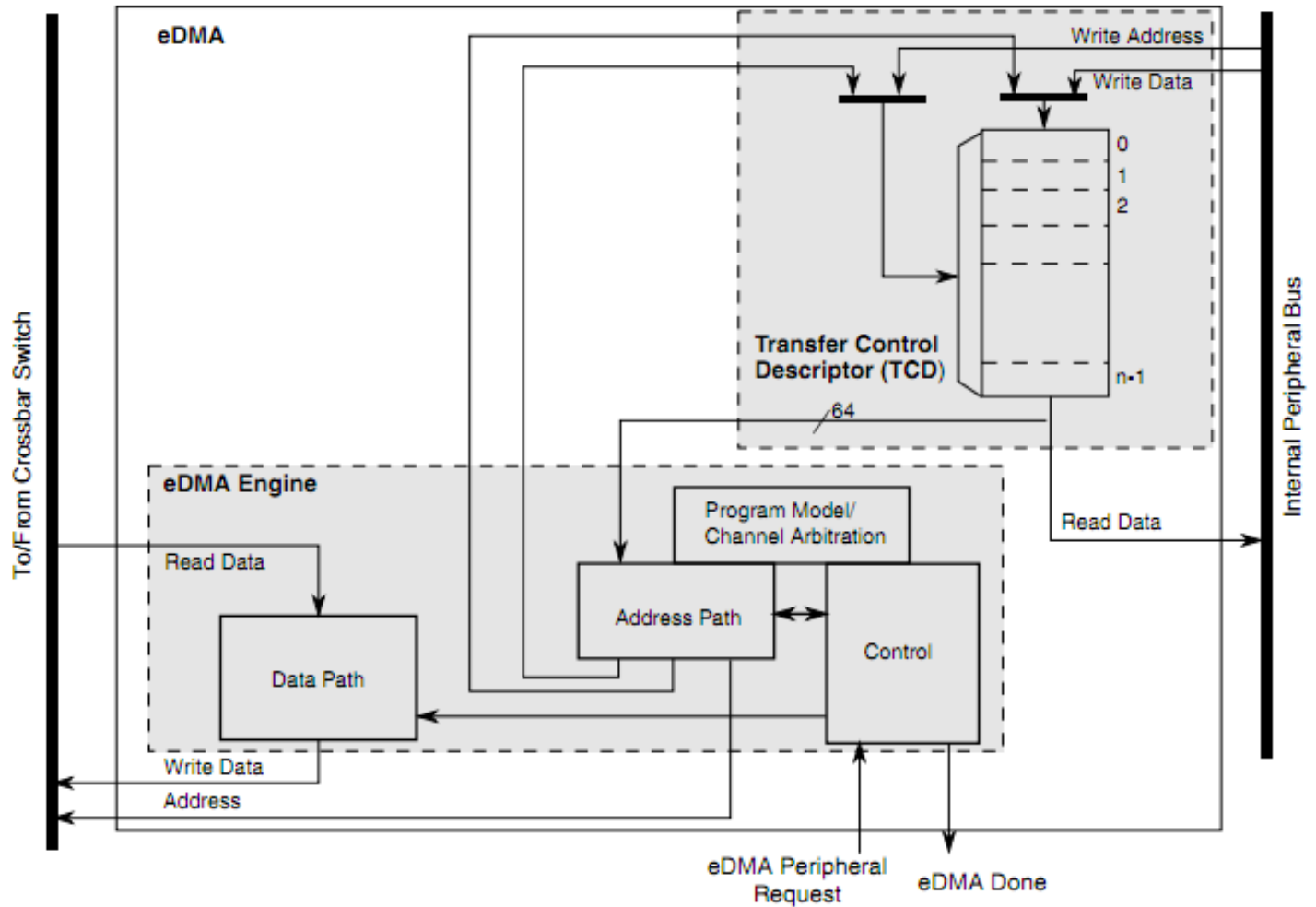
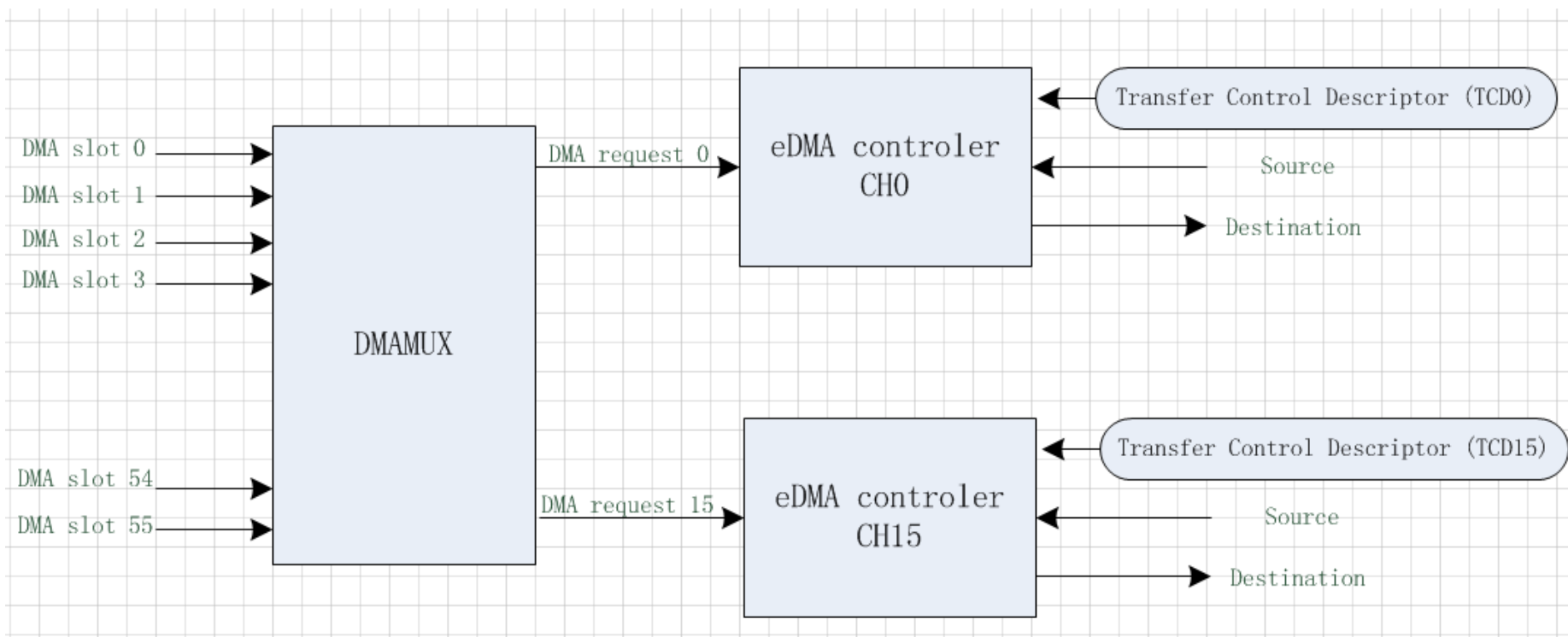


Figure 21-1. eDMA block diagram



# DMA控制器在CMOS图像读出中的应用

- 为了方便理解，做了DMA工作的简化框图





# DMA控制器在CMOS图像读出中的应用

---

- TCD(Transfer Control Descriptor)是控制DMA功能的核心部件，它描述从源地址到目标地址该如何传输。
- 在K10中，TCD由15个寄存器构成。
- TCD的寄存器中，有以下几种  
SADDR SOFF ATTR NBYTES SLAST  
DADDR DOFF CITER DLAST CSR BITER
- 下面讲几个主要的寄存器

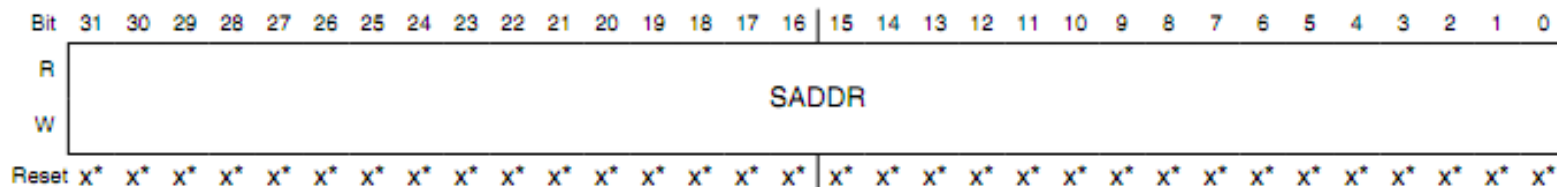


# DMA控制器在CMOS图像读出中的应用

## ■ SADDR DADDR 源地址、目标地址

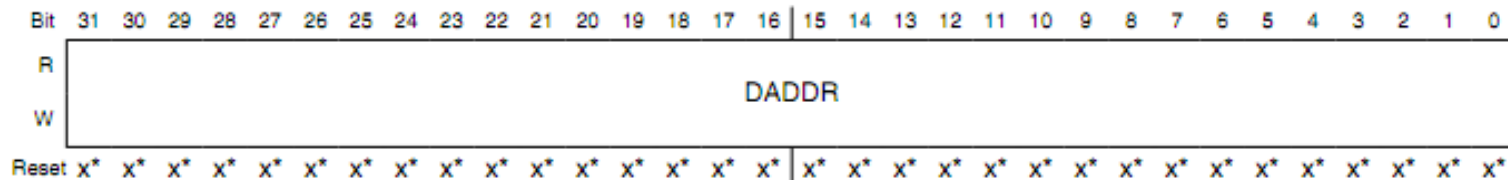
### 21.3.17 TCD Source Address (DMA\_TCDn\_SADDR)

Addresses: 4000\_8000h base + 1000h offset + (32d × n), where n = 0d to 15d



### 21.3.24 TCD Destination Address (DMA\_TCDn\_DADDR)

Addresses: 4000\_8000h base + 1010h offset + (32d × n), where n = 0d to 15d

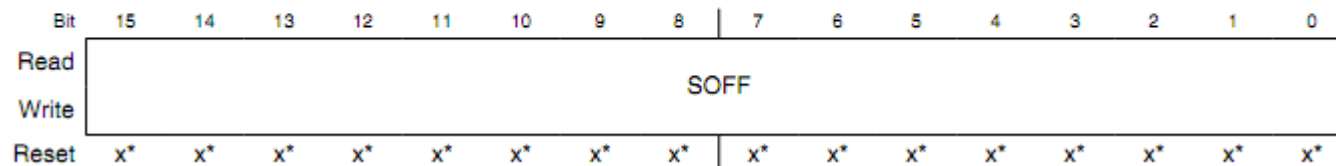


# DMA控制器在CMOS图像读出中的应用

## ■ SOFF DOFF 源偏移、目标偏移

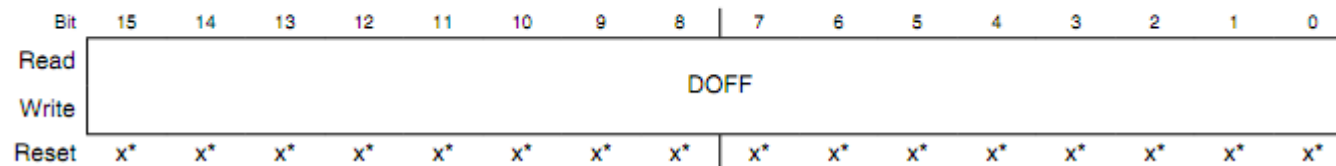
### 21.3.18 TCD Signed Source Address Offset (DMA\_TCDn\_SOFF)

Addresses: 4000\_8000h base + 1004h offset + (32d × n), where n = 0d to 15d



### 21.3.25 TCD Signed Destination Address Offset (DMA\_TCDn\_DOFF)

Addresses: 4000\_8000h base + 1014h offset + (32d × n), where n = 0d to 15d



# DMA控制器在CMOS图像读出中的应用

- SSIZE DSIZE 源大小、目标大小
- SMOD DMOD 源地址模数、目标地址模数

## 21.3.19 TCD Transfer Attributes (DMA\_TCDn\_ATTR)

Addresses: 4000\_8000h base + 1006h offset + (32d × n), where n = 0d to 15d

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
Read	SMOD								SSIZE				DMOD				DSIZE	
Write																		
Reset	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*		



# DMA控制器在CMOS图像读出中的应用

---

## ■ NBYTES 次要字节传输计数

21.3.20 TCD Minor Byte Count (Minor Loop Disabled)  
(DMA\_TCDn\_NBYTES\_MLNO)

21.3.21 TCD Signed Minor Loop Offset (Minor Loop Enabled and  
Offset Disabled) (DMA\_TCDn\_NBYTES\_MLOFFNO)

21.3.22 TCD Signed Minor Loop Offset (Minor Loop and Offset  
Enabled) (DMA\_TCDn\_NBYTES\_MLOFFYES)



# DMA控制器在CMOS图像读出中的应用

0x40000000	0x00	0x40000000	0x00
0x40000001	0x01	0x40000001	0x01
0x40000002	0x02	0x40000002	0x02
0x40000003	0x03	0x40000003	0x03
0x40000004	0x04	0x40000004	0x04
0x40000005	0x05	0x40000005	0x05

DMA request



0x40001000	0x00	0x40001000	0x00
0x40001001	0x00	0x40001001	0x01
0x40001002	0x00	0x40001002	0x02
0x40001003	0x00	0x40001003	0x03
0x40001004	0x00	0x40001004	0x04
0x40001005	0x00	0x40001005	0x05

SADDR=0x40000000,DADDR=0x40001000

SOFF=1,DOFF=1

SIZE=8bit,DSIZE=8bit

NBYTES=6



# DMA控制器在CMOS图像读出中的应用

0x40000000	0x00	0x40000000	0x00
0x40000001	0x01	0x40000001	0x01
0x40000002	0x02	0x40000002	0x02
0x40000003	0x03	0x40000003	0x03
0x40000004	0x04	0x40000004	0x04
0x40000005	0x05	0x40000005	0x05

DMA request



0x40001000	0x00	0x40001000	0x00
0x40001001	0x00	0x40001001	0x02
0x40001002	0x00	0x40001002	0x04
0x40001003	0x00	0x40001003	0x00
0x40001004	0x00	0x40001004	0x00
0x40001005	0x00	0x40001005	0x00

SADDR=0x40000000,DADDR=0x40001000

SOFF=2,DOFF=1

SIZE=8bit,DSIZE=8bit

NBYTES=3



# DMA控制器在CMOS图像读出中的应用

---

## ■ CITER BITER 当前迭代计数、主迭代计数起始值

21.3.26 TCD Current Minor Loop Link, Major Loop Count (Channel Linking Enabled) (DMA\_TCDn\_CITER\_ELINKYES)

21.3.27 TCD Current Minor Loop Link, Major Loop Count (Channel Linking Disabled) (DMA\_TCDn\_CITER\_ELINKNO)

21.3.30 TCD Beginning Minor Loop Link, Major Loop Count (Channel Linking Enabled) (DMA\_TCDn\_BITER\_ELINKYES)

21.3.31 TCD Beginning Minor Loop Link, Major Loop Count (Channel Linking Disabled) (DMA\_TCDn\_BITER\_ELINKNO)



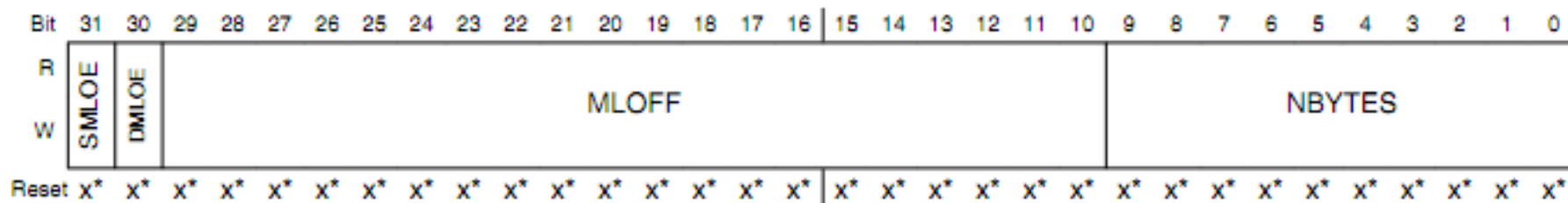


# DMA控制器在CMOS图像读出中的应用

## ■ MLOFF 次要循环偏移

### 21.3.22 TCD Signed Minor Loop Offset (Minor Loop and Offset Enabled) (DMA\_TCDn\_NBYTES\_MLOFFYES)

Addresses: 4000\_8000h base + 1008h offset + (32d x n), where n = 0d to 15d

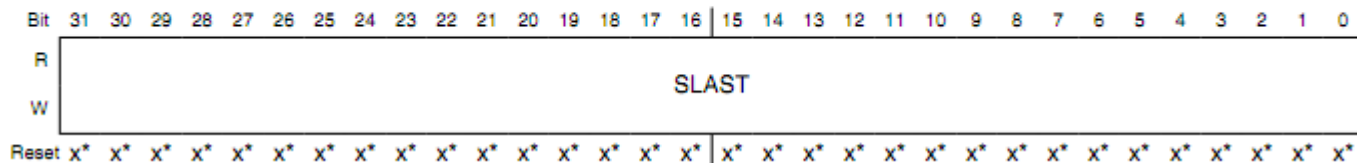


# DMA控制器在CMOS图像读出中的应用

## ■ SLAST DLAST 末级源地址调整值，末级目标地址调整值

### 21.3.23 TCD Last Source Address Adjustment (DMA\_TCDn\_SLAST)

Addresses: 4000\_8000h base + 100Ch offset + (32d × n), where n = 0d to 15d



### 21.3.28 TCD Last Destination Address Adjustment/Scatter Gather Address (DMA\_TCDn\_DLASTSGA)

Addresses: 4000\_8000h base + 1018h offset + (32d × n), where n = 0d to 15d



# DMA控制器在CMOS图像读出中的应用

## ■ CSR 传输状态与控制

### 21.3.29 TCD Control and Status (DMA\_TCDn\_CSR)

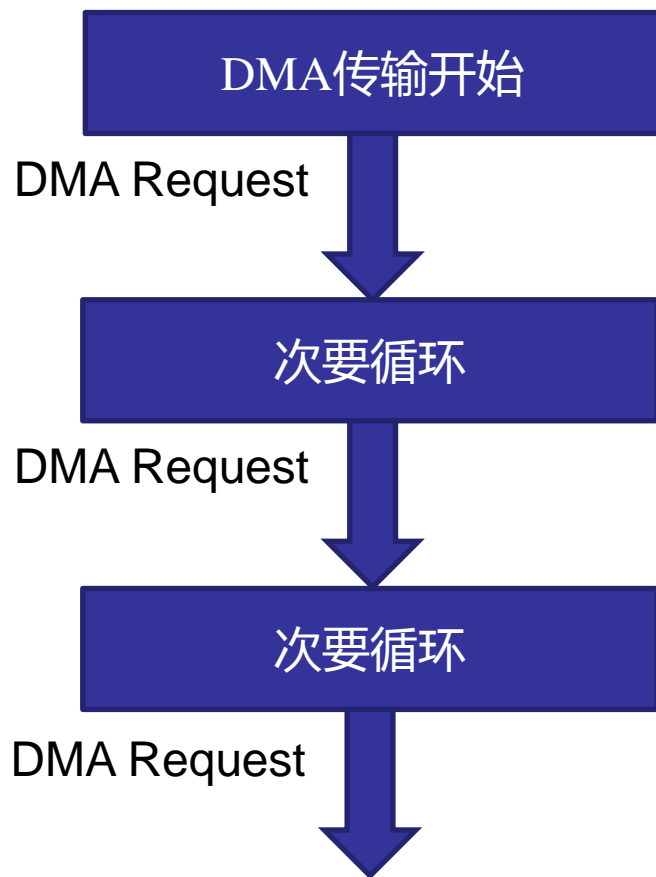
Addresses: 4000\_8000h base + 101Ch offset + (32d × n), where n = 0d to 15d

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Read	BWC		0		MAJORLINKCH				DONE	ACTIVE	MAJORELINK	ESG	DREQ	INTHALF	INTMAJOR	START
Write	BWC		0		MAJORLINKCH				DONE	ACTIVE	MAJORELINK	ESG	DREQ	INTHALF	INTMAJOR	START
Reset	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*



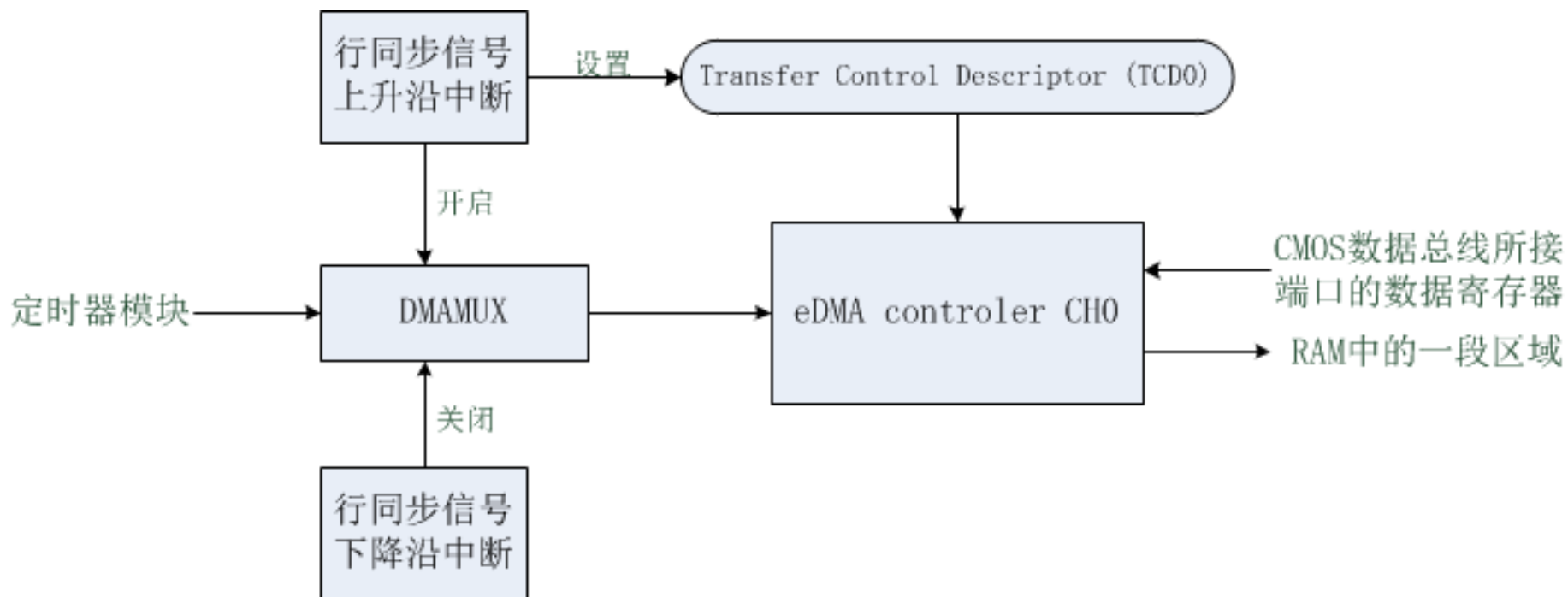
# DMA控制器在CMOS图像读出中的应用

---



# DMA控制器在CMOS图像读出中的应用

## ■ 用DMA模式读摄像头数据的结构框图



# DMA控制器在CMOS图像读出中的应用

---

- 实测结果
- 在64MHz的CPU时钟下，摄像头以60Hz输出的每帧图像都能采得，一帧图像采集 $240 \times 160$ 个点，即采下了所有行，每行采集160个点，总共占用38.4kbyte内存空间。
- 完成图像采集工作，CPU占用率仅有11%



# K10的DSP功能的应用

---

- K10使用ARM Cortex-M4的核，具有1.25DMIPS/MHz的运算能力，突出特点是加入了一些支持数字信号处理（DSP）的指令。
- Cortex-M4核具有一个单时钟周期乘法累加（MAC）单元、优化的单指令多数据（SIMD）指令、饱和运算指令和一个可选的单精度浮点运算单元（FPU）。
- 单时钟周期乘法累加（MAC）单元可以大大提高FFT变换及数字滤波的执行效率。
- 在用DMA进行图像采集节省大量CPU时间之后，K10的M4的核使进行复杂的图像处理算法变得有可能。



# K10的DSP功能的应用

---

- 如何使用Cortex-M4的核应该是编译器的事情。
- 在IAR6.21(IAR Embedded Workbench IDE)软件下找到了专门为ARM Cortex-M4设计的数字信号处理的库，使用这个库里的函数，编译器在编译时会根据M4的核进行优化。
- 库的位置在\IARARM\arm\CMSIS\DSP\_Lib\Source\IAR
- 提供了16位、32位整数、浮点数的一些复杂运算，包括FIR 滤波器、FFT、反FFT等函数。





---

---

谢谢~

■ 欢迎提问~

