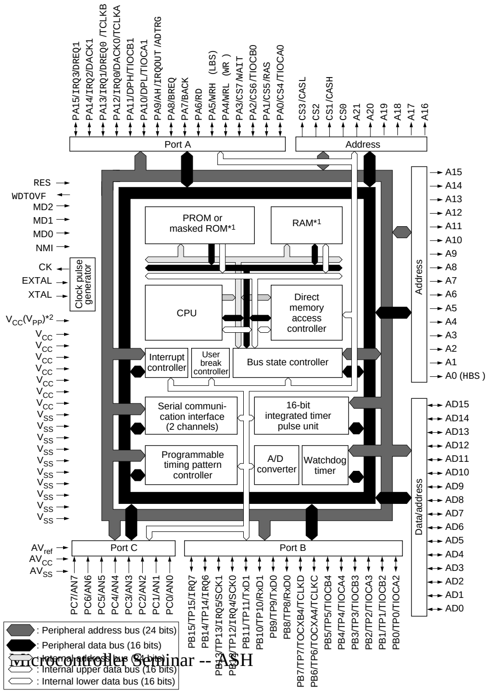
Advanced Embedded Microcontroller Seminar--Day 2

Andrew Huang
bunnie@mit.edu

Agenda

- Day 2: 1/21
  - Designing the SH-1 into your system
  - Details of the SH1 integrated peripherals
  - Overview of the SH1WH, on-board peripheral specs
  - development environment

SH-1 Block Diagram



Location of Hardware Manual

- http://semiconductor.hitachi.com/products/h_micon/1_sh/1_sh1/H11TH003D3/pdf/h1103.pdf
  - difficult to read sometimes, but the comprehensive source of information on the SH-1
  - today’s lecture distills and presents some of the most important concepts to get you started

Getting in to the SH-1: Features

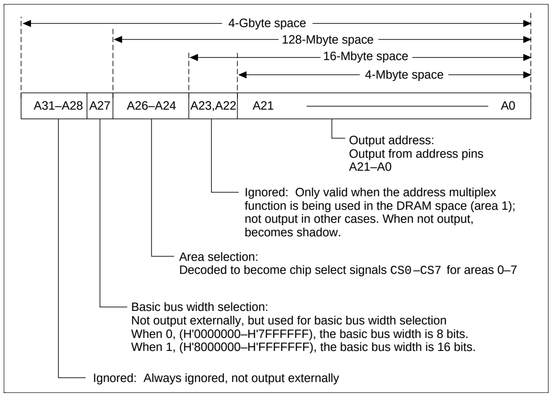
- 32-bit RISC CPU with MAC instruction
- interrupt controller
  - 9 external interrupt pins
  - 31 internal interrupt sources
  - 16 priority levels
- user break controller
  - for trapping on special conditions, debugging
- clock generator
  - drives an external crystal to make internal timebase
- bus state controller
  - 8/16 bit accesses
  - muxed or separate address and data
  - controls DRAM, SRAM, ROM, or peripherals
  - generates up to 8 chip selects
  - wait states
- DMA controller
  - 4 channels, 2 internal, 2 external
  - transfer between external and internal memory, I/O
  - cycle-steal and burst-mode

- 16-bit integrated timer unit
  - measures input pulse widths
  - generates PWM, complementary PWM
  - phase counting mode
- timing pattern controller
  - can be used in conjunction with DMAC and ITU to generate arbitrary pulse trains with no load on CPU
  - non-overlapping mode available
  - applications include multi-phase motor control, serial communications (SPI, microwire, etc.)
- watchdog timer
- serial comm. interface
  - 2 channels
  - async/synch options
  - internal baud rate generator
- A/D converter
  - 10 bits, 8 channels
  - external trigger, reference source
- digital I/O ports
  - up to 40 lines, but all shared with other I/O functions
- on-chip RAM/ROM
  - SH7032 (ROM-less) has 8K RAM
  - SH7034 has 64K PROM plus 4K RAM
  - single-cycle access to on-chip RAM/ROM

SH-1: Key Design Point

- All peripherals and memory are mapped to one common address space
  - 28-bit effective address space is divided into 16 equally sized sectors
    - 28 bits because top 4 bits of a 32-bit address are don’t care
  - actual usage of allocated space per sector varies; undesignated areas alias to designated areas
  - processor bus behavior (ie, 8 or 16 bit mode, wait states, multiplexed adr, data) determined by which address is accessed
    - details of behavior can be modified by setting memory-mapped registers in the BSC
- Vector table is located at bottom of memory
  - hence there must always be a ROM mapped to CS0

Memory organization of the SH-1



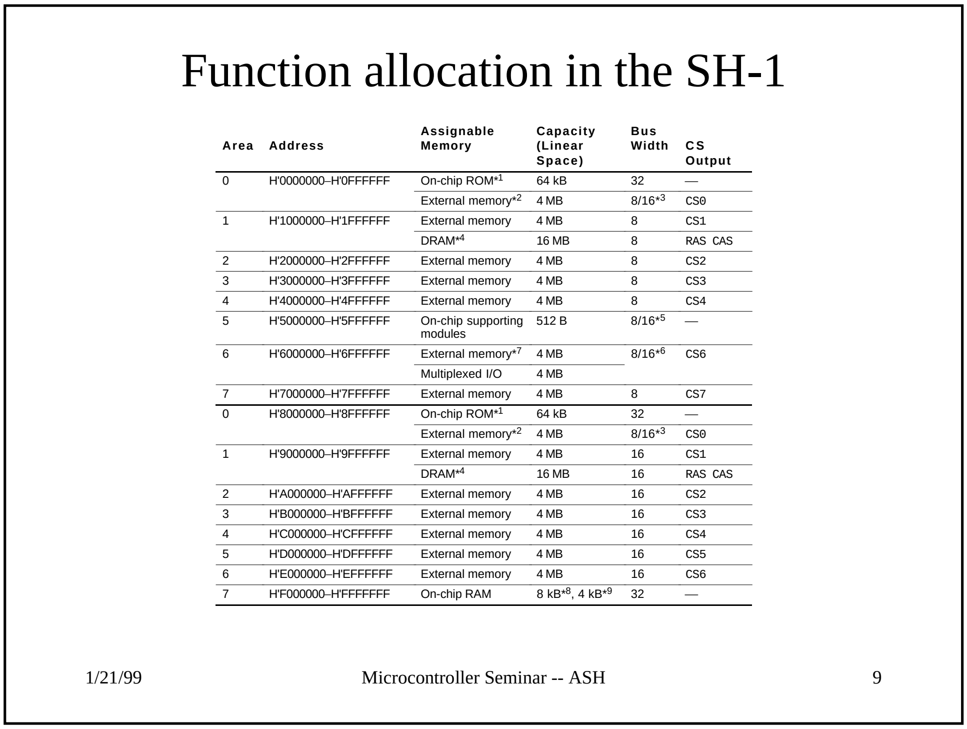
Output address:
Output from address pins
A21–A0

Ignored: Only valid when the address multiplex function is being used in the DRAM space (area 1); not output in other cases. When not output, becomes shadow.

Area selection:
Decoded to become chip select signals CS0–CS7 for areas 0–7

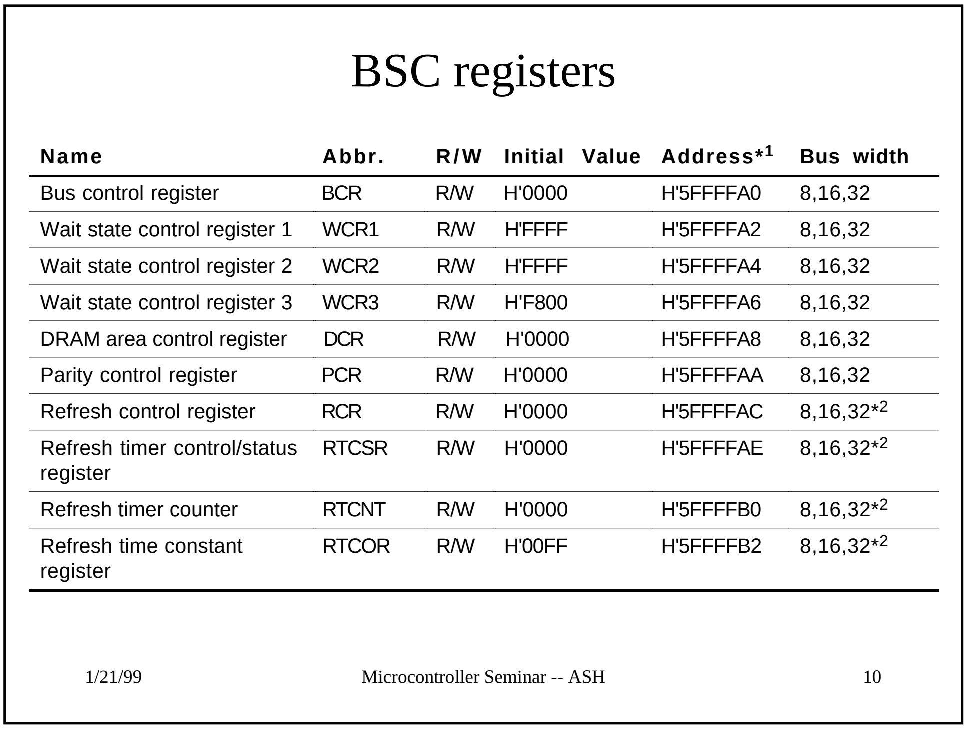
Basic bus width selection:
Not output externally, but used for basic bus width selection
When 0, (H'0000000–H'7FFFFFF), the basic bus width is 8 bits.
When 1, (H'8000000–H'FFFFFFF), the basic bus width is 16 bits.

Ignored: Always ignored, not output externally



Function allocation in the SH-1

Area	Address	Assignable Memory	Capacity (Linear Space)	Bus Width	CS Output
0	H'0000000–H'0FFFFFF	On-chip ROM*[1]	64 kB	32	—
		External memory*[2]	4 MB	8/16*[3]	CS0
1	H'1000000–H'1FFFFFF	External memory	4 MB	8	CS1
		DRAM*[4]	16 MB	8	RAS CAS
2	H'2000000–H'2FFFFFF	External memory	4 MB	8	CS2
3	H'3000000–H'3FFFFFF	External memory	4 MB	8	CS3
4	H'4000000–H'4FFFFFF	External memory	4 MB	8	CS4
5	H'5000000–H'5FFFFFF	On-chip supporting modules	512 B	8/16*[5]	—
6	H'6000000–H'6FFFFFF	External memory*[7]	4 MB	8/16*[6]	CS6
		Multiplexed I/O	4 MB		
7	H'7000000–H'7FFFFFF	External memory	4 MB	8	CS7
0	H'8000000–H'8FFFFFF	On-chip ROM*[1]	64 kB	32	—
		External memory*[2]	4 MB	8/16*[3]	CS0
1	H'9000000–H'9FFFFFF	External memory	4 MB	16	CS1
		DRAM*[4]	16 MB	16	RAS CAS
2	H'A000000–H'AFFFFFF	External memory	4 MB	16	CS2
3	H'B000000–H'BFFFFFF	External memory	4 MB	16	CS3
4	H'C000000–H'CFFFFFF	External memory	4 MB	16	CS4
5	H'D000000–H'DFFFFFF	External memory	4 MB	16	CS5
6	H'E000000–H'EFFFFFF	External memory	4 MB	16	CS6
7	H'F000000–H'FFFFFFF	On-chip RAM	8 kB*[8], 4 kB*[9]	32	—



BSC registers

Name	Abbr.	R/W	Initial Value	Address*[1]	Bus width
Bus control register	BCR	R/W	H'0000	H'5FFFFA0	8,16,32
Wait state control register 1	WCR1	R/W	H'FFFF	H'5FFFFA2	8,16,32
Wait state control register 2	WCR2	R/W	H'FFFF	H'5FFFFA4	8,16,32
Wait state control register 3	WCR3	R/W	H'F800	H'5FFFFA6	8,16,32
DRAM area control register	DCR	R/W	H'0000	H'5FFFFA8	8,16,32
Parity control register	PCR	R/W	H'0000	H'5FFFFAA	8,16,32
Refresh control register	RCR	R/W	H'0000	H'5FFFFAC	8,16,32*[2]
Refresh timer control/status register	RTCSR	R/W	H'0000	H'5FFFFAE	8,16,32*[2]
Refresh timer counter	RTCNT	R/W	H'0000	H'5FFFFB0	8,16,32*[2]
Refresh time constant register	RTCOR	R/W	H'00FF	H'5FFFFB2	8,16,32*[2]

Registers of the BSC

- BCR
  - determines basic signal configuration (DRAM mode on area 1, mux’d adr/data on area 6, warp mode, RD# duty cycle, byte access style
- WCR[1:3]
  - configures wait states for normal, DMA accesses
- DCR
  - configures basic DRAM access timing (dual CAS, precharge time, FPM, CAS duty cycle)
- RCR
  - configures DRAM refresh timing (CBR, self-refresh, wait states)
- RTCSR, RTCNT
  - configures refresh period based on system clock
- PCR
  - configures parity checking modes

PFC

- Pin function controller
  - determines functions of multiplexed pins
  - almost every I/O pin has multiple functions
  - must initialize to select proper mode prior to using any peripheral function
  - many basic services, such as some chip select lines, RD#, WR#, CAS/RAS can be turned on or off by writing to this register
    - fastest way to wedge the system is to write the wrong info to these registers

PFC registers

Name	Abbreviation	R/W	Initial Value	Address	Access Size
Port A I/O register	PAIOR	R/W	H'0000	H'5FFFFC4	8, 16, 32
Port A control register 1	PACR1	R/W	H'3302	H'5FFFFC8	8, 16, 32
Port A control register 2	PACR2	R/W	H'FF95	H'5FFFFCA	8, 16, 32
Port B I/O register	PBIOR	R/W	H'0000	H'5FFFFC6	8, 16, 32
Port B control register 1	PBCR1	R/W	H'0000	H'5FFFFCC	8, 16, 32
Port B control register 2	PBCR2	R/W	H'0000	H'5FFFFCE	8, 16, 32
Column address strobe pin control register	CASCR	R/W	H'5FFF	H'5FFFFEE	8, 16, 32

- P{A,B}CR{1,2} configures which mux'd function to use
- P{A,B}IOR configures, on a pin by pin basis, input or output mode if digital I/O mode is selected
- CASCR determines function of mux'd CS1,3 and CASL,H lines

Digital I/O

- Referred to as parallel I/O in the hardware manual
  - any of port A or B can be configured to be input or output
  - port C is input only
  - ports A and B are each 16 bits wide at max, port C is 8 bits wide at max
  - port C is mux'd with analog input, and automagically functions as analog input when A/D is activated
- PADR, PBDR, and PCDR are R/W registers that will either reflect the data on the pin, or contain the data to be driven on pin, depending on pin configuration

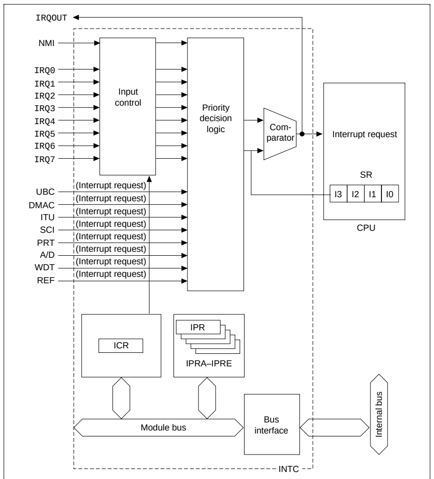
Interrupt Controller

- 16 priority levels
- up to 255 interrupt vectors
  - about 43 are actually assigned to interrupt sources
  - vector table starts at memory location 0, each vector consisting of a 32-bit pointer to code to handle the interrupt
  - power on reset and manual reset each has two vectors, the first for setting the PC, the second for setting the SP
- 8 external interrupt inputs
- IRQOUT line to indicate presence of internal interrupts
- edge or level triggered, programmable by software

Interrupts

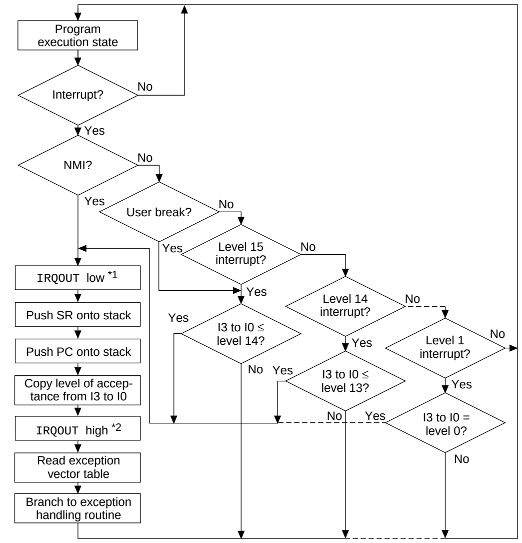
- on interrupt,
  - status register is pushed (32-bit)
  - return PC is pushed (32-bit)
  - note that SP should always be a multiple of 4
- interrupt response latency
  - latency dependant upon how fast long word accesses to memory occur
  - min. latency is 11 cycles (8 cycles for internal pipes to flush, plus 3 cycles for vector fetch and stack pushes)
  - max. latency is 12 + 2(SR, PC, vector access times) + fetch time for first handler instruction
  - typ. latency around 1 to 2 us using slow memory at 12 MHz
- see table 4.10 in hardware manual for precise behavior of machine on a wide range of conditions

Interrupt Controller Hardware

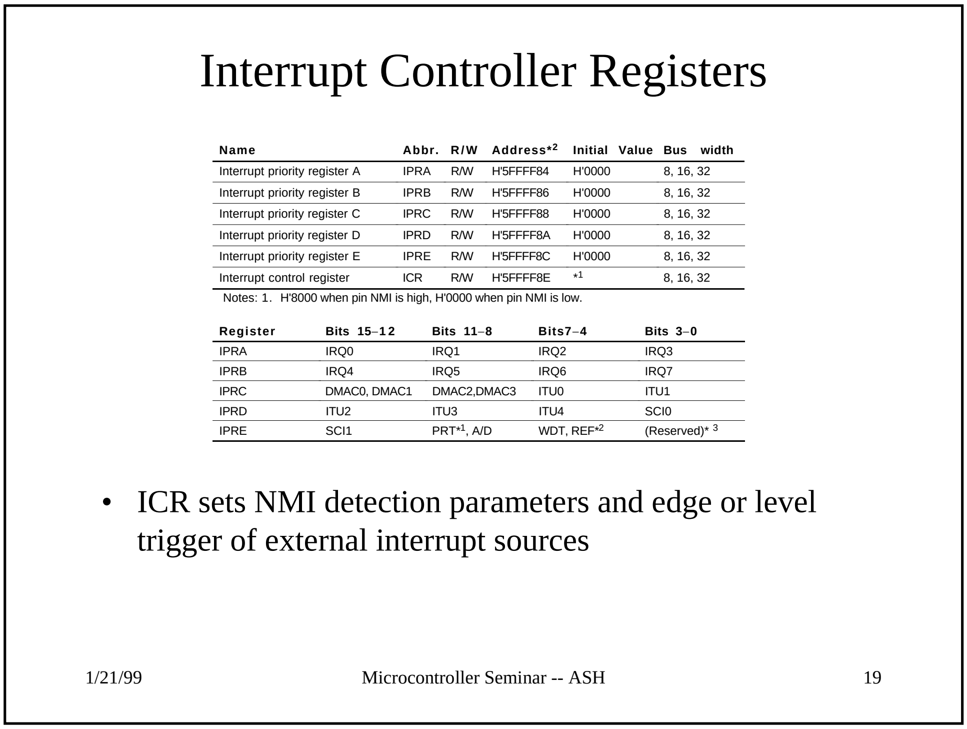


UBC: User break controller WDT: Watchdog timer
DMAC: Direct memory access controller REF: DRAM refresh control unit of BSC
ITU: 16-bit integrated timer pulse unit ICR: Interrupt control register
SCI: Serial communication interface IPRA–IPRE: Interrupt priority registers A–E
PRT: Parity control unit of BSC SR: Status register
A/D: A/D converter

Interrupt Processing Flow Chart



I3 to I0 : Interrupt mask bits of status register



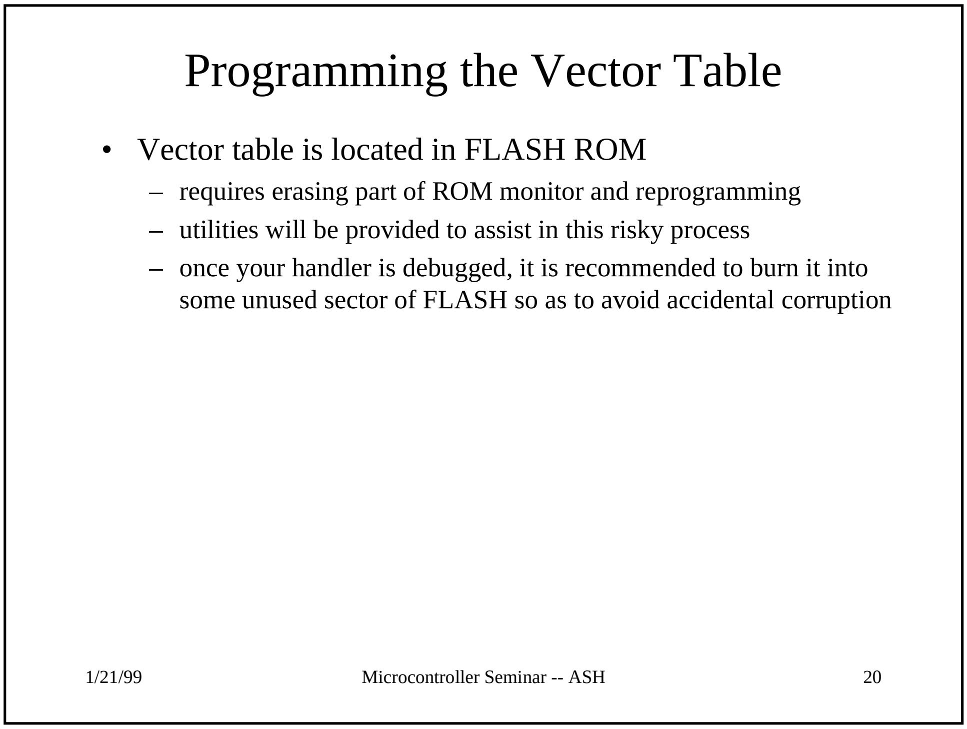
Interrupt Controller Registers

Name	Abbr.	R/W	Address*2	Initial Value	Bus width
Interrupt priority register A	IPRA	R/W	H'5FFFF84	H'0000	8, 16, 32
Interrupt priority register B	IPRB	R/W	H'5FFFF86	H'0000	8, 16, 32
Interrupt priority register C	IPRC	R/W	H'5FFFF88	H'0000	8, 16, 32
Interrupt priority register D	IPRD	R/W	H'5FFFF8A	H'0000	8, 16, 32
Interrupt priority register E	IPRE	R/W	H'5FFFF8C	H'0000	8, 16, 32
Interrupt control register	ICR	R/W	H'5FFFF8E	*1	8, 16, 32

Notes: 1. H'8000 when pin NMI is high, H'0000 when pin NMI is low.

Register	Bits 15–12	Bits 11–8	Bits7–4	Bits 3–0
IPRA	IRQ0	IRQ1	IRQ2	IRQ3
IPRB	IRQ4	IRQ5	IRQ6	IRQ7
IPRC	DMAC0, DMAC1	DMAC2,DMAC3	ITU0	ITU1
IPRD	ITU2	ITU3	ITU4	SCI0
IPRE	SCI1	PRT*1, A/D	WDT, REF*2	(Reserved)* 3

- ICR sets NMI detection parameters and edge or level trigger of external interrupt sources



Programming the Vector Table

- Vector table is located in FLASH ROM
  - requires erasing part of ROM monitor and reprogramming
  - utilities will be provided to assist in this risky process
  - once your handler is debugged, it is recommended to burn it into some unused sector of FLASH so as to avoid accidental corruption

Integrated Timer Pulse Unit

Item	Channel 0	Channel 1	Channel 2	Channel 3	Channel 4
Counter clocks	Internal: ϕ, ϕ/2, ϕ/4,ϕ/8 External: Independently selectable from TCLKA, TCLKB, TCLKC, and TCLKD				
General registers (output compare/ input capture dual registers)	GRA0, GRB0	GRA1, GRB1	GRA2, GRB2	GRA3, GRB3	GRA4, GRB4
Buffer registers	No	No	No	BRA3, BRB3	BRA4, BRB4
Input/output pins	TIOCA0, TIOCB0	TIOCA1, TIOCB1	TIOCA2, TIOCB2	TIOCA3, TIOCB3	TIOCA4, TIOCB4
Output pins	No	No	No	No	TOCXA4, TOCXB4
Counter clear function (compare match or input capture)	GRA0/GRB0	GRA1/GRB1	GRA2/GRB2	GRA3/GRB3	GRA4/GRB4
0	Yes	Yes	Yes	Yes	Yes
1	Yes	Yes	Yes	Yes	Yes
Toggle output	Yes	Yes	No	Yes	Yes
Input capture function	Yes	Yes	Yes	Yes	Yes
Synchronization	Yes	Yes	Yes	Yes	Yes
PWM mode	Yes	Yes	Yes	Yes	Yes
Reset-synchronized PWM mode	No	No	No	Yes	Yes
Complementary PWM mode	No	No	No	Yes	Yes
Phase counting mode	No	No	Yes	No	No
Buffer operation	No	No	No	Yes	Yes
DMAC activation	GRA0 compare match or input capture	GRA1 compare match or input capture	GRA2 compare match or input capture	GRA3 compare match or input capture	No
Interrupt sources (three)	• Compare match/input capture A0 • Compare match/input capture B0 • Overflow	• Compare match/input capture A1 • Compare match/input capture B1 • Overflow	• Compare match/input capture A2 • Compare match/input capture B2 • Overflow	• Compare match/input capture A3 • Compare match/input capture B3 • Overflow	• Compare match/input capture A4 • Compare match/input capture B4 • Overflow

ITU Registers

Channel	Name	Abbreviation	R/W	Initial Value	Address*[1]	Access Size
Shared	Timer start register	TSTR	R/W	H'E0/H'60	H'5FFFF00	8
	Timer synchro register	TSNC	R/W	H'E0/H'60	H'5FFFF01	8
	Timer mode register	TMDR	R/W	H'80/H'00	H'5FFFF02	8
	Timer function control register	TFCR	R/W	H'C0/H'40	H'5FFFF03	8
	Timer output control register	TOCR	R/W	H'FF/H'7F	H'5FFFF31	8
3	Timer control register 3	TCR3	R/W	H'80/H'00	H'5FFFF22	8
	Timer I/O control register 3	TIOR3	R/W	H'88/H'08	H'5FFFF23	8
	Timer interrupt enable register 3	TIER3	R/W	H'F8/H'78	H'5FFFF24	8
	Timer status register 3	TSR3	R/(W)*[2]	H'F8/H'78	H'5FFFF25	8
	Timer counter 3	TCNT3	R/W	H'00	H'5FFFF26	8, 16
					H'5FFFF27	8, 16
	General register A3	GRA3	R/W	H'FF	H'5FFFF28	8, 16, 32
					H'5FFFF29	8, 16, 32
	General register B3	GRB3	R/W	H'FF	H'5FFFF2A	8, 16, 32
					H'5FFFF2B	8, 16, 32
	Buffer register A3	BRA3	R/W	H'FF	H'5FFFF2C	8, 16, 32
					H'5FFFF2D	8, 16, 32
	Buffer register B3	BRB3	R/W	H'FF	H'5FFFF2E	8, 16, 32
					H'5FFFF2F	8, 16, 32

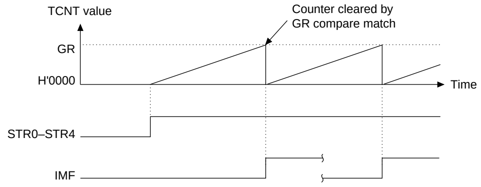
ITU Register Cross-Reference Matrix

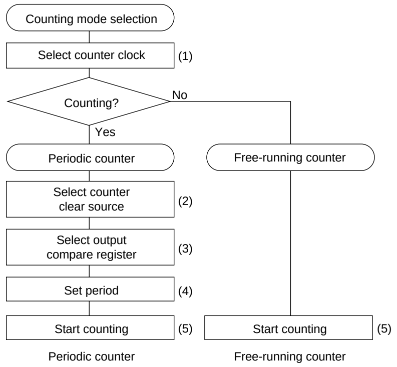
See tables at end of ITU section in hardware manual for complete matrix

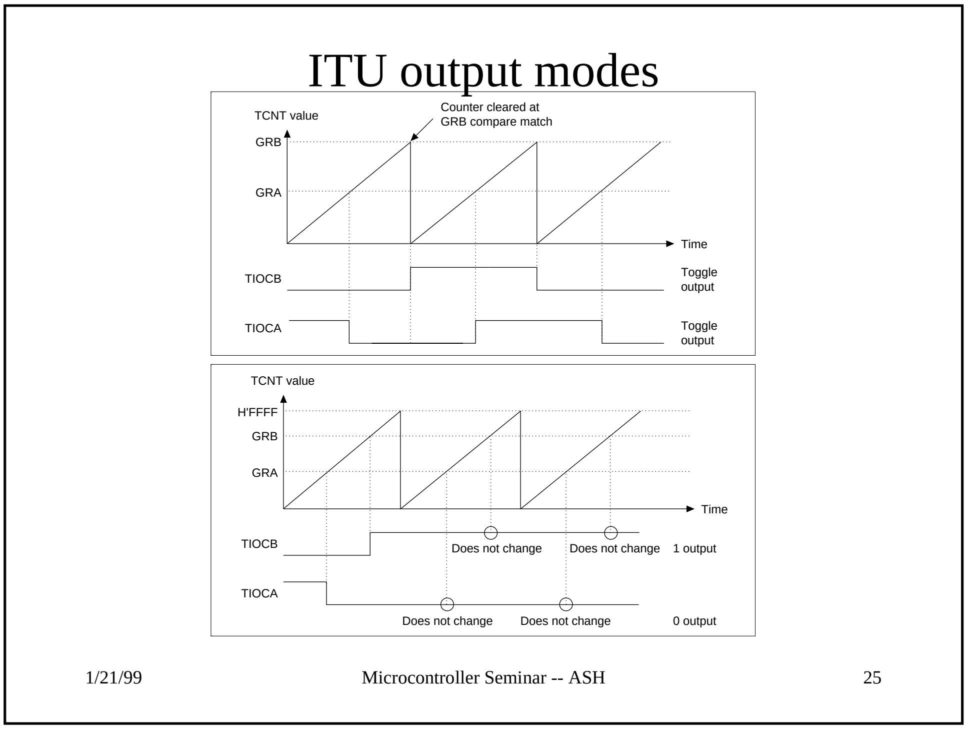
	Register Setting											
	TSNC	TMDR			TFCR			TOCR	TIOR0		TCR0	
Operating Mode	Sync	MDF	FDIR	PWM	Comp PWM	Reset Sync PWM	Buffer	Output Level Select	IOA	IOB	Clear Select	Clock Select
Synch-ronized preset	SYNC0 =1	—	—	√	—	—	—	—	√	√	√	√
PWM	√	—	—	PWM0 =1	—	—	—	—	—	√	√	√
Output compare A function	√	—	—	PWM0 =0	—	—	—	—	IOA2 = 0, others: don't care	√	√	√
Output compare B function	√	—	—	√	—	—	—	—	√	IOB2 = 0, others : don't care	√	√
Input capture A function	√	—	—	PWM0 =0	—	—	—	—	IOA2 = 1, others: don't care	√	√	√
Input capture B function	√	—	—	PWM0 =0	—	—	—	—	√	IOB2 = 1, others : don't care	√	√
Counter Clear Function												
Clear at compare match/input capture A	√	—	—	√	—	—	—	—	√	√	CCLR1 = 0 CCLR0 =1	√
Clear at compare match/input capture B	√	—	—	√	—	—	—	—	√	√	CCLR1 = 1 CCLR0 =0	√
Synch-ronized clear	SYNC0 =1	—	—	√	—	—	—	—	√	√	CCLR1 = 1 CCLR0 =1	√

√: Settable, —: Setting does not affect current mode

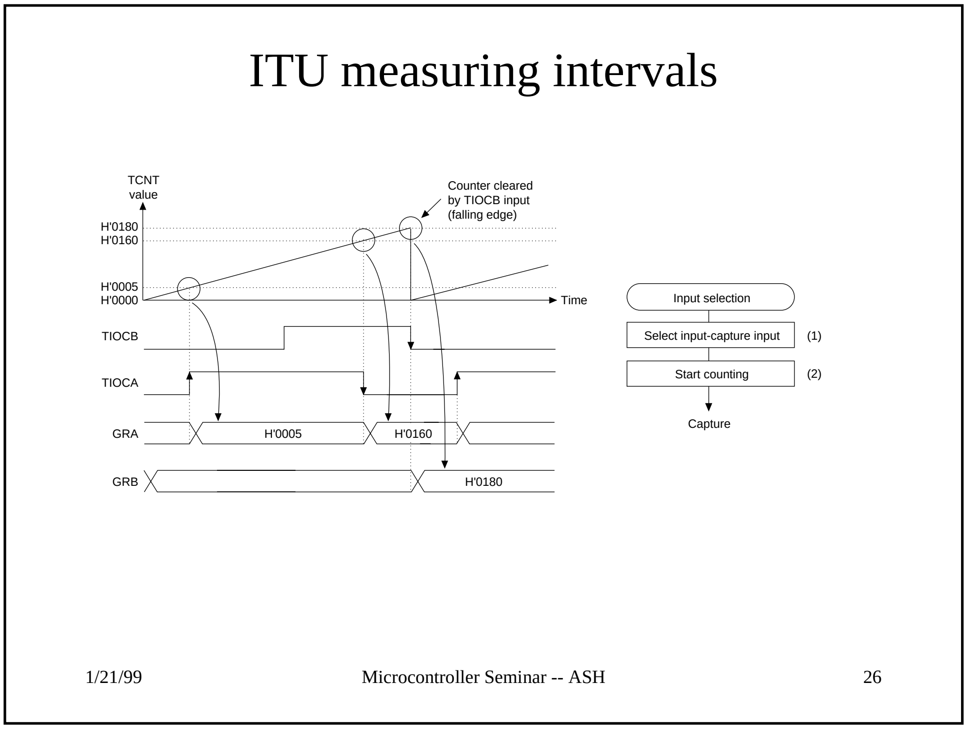
Simple ITU Example







ITU output modes



ITU measuring intervals

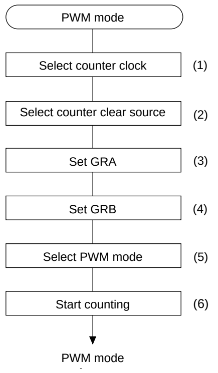
Input selection

Select input-capture input (1)

Start counting (2)

Capture

ITU in PWM mode


PWM mode

Select counter clock (1)

Select counter clear source (2)

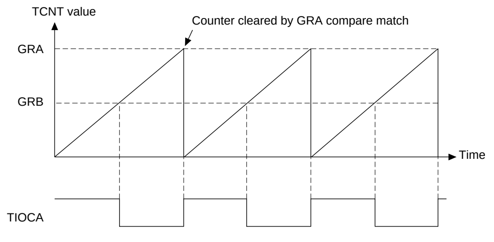
Set GRA (3)

Set GRB (4)

Select PWM mode (5)

Start counting (6)

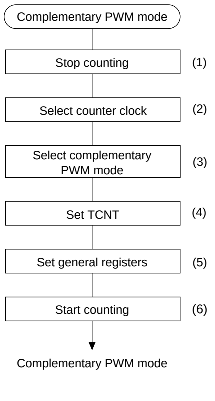
PWM mode



a. Counter cleared by GRA

ITU complementary PWM mode

- Non-overlapping complementary PWM outputs


Complementary PWM mode

Stop counting (1)

Select counter clock (2)

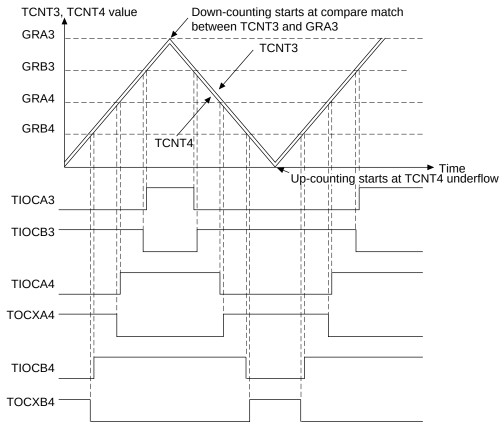
Select complementary PWM mode (3)

Set TCNT (4)

Set general registers (5)

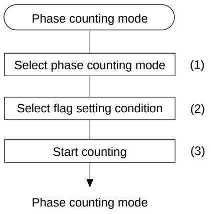
Start counting (6)

Complementary PWM mode


TCNT3, TCNT4 value
Down-counting starts at compare match between TCNT3 and GRA3
GRA3
TCNT3
GRB3
GRA4
GRB4
TCNT4
Time
Up-counting starts at TCNT4 underflow
TIOCA3
TIOCB3
TIOCA4
TOCXA4
TIOCB4
TOCXB4

ITU in phase counting mode

- Use quadrature to figure out direction of rotation in shaft encoders



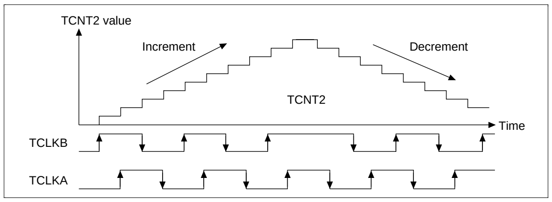
Phase counting mode

Select phase counting mode (1)

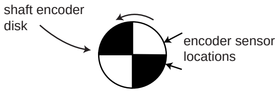
Select flag setting condition (2)

Start counting (3)

Phase counting mode



Counting Direction	Increment				Decrement			
TCLKB	Rising	High	Falling	Low	Rising	High	Falling	Low
TCLKA	Low	Rising	High	Falling	High	Falling	Low	Rising

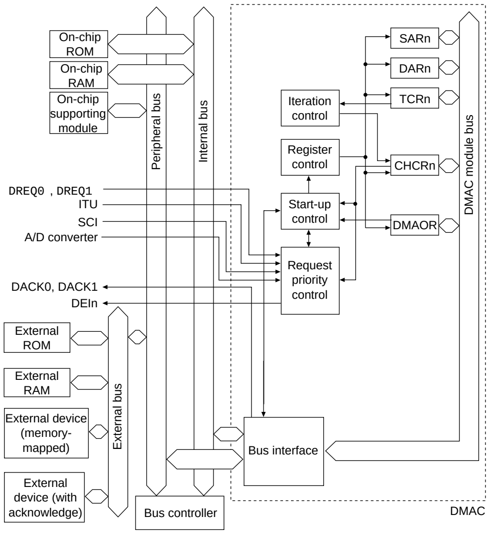


Direct Memory Access Controller

- 4-channel
  - 2 internal
  - 2 external
- 64K transfers at a time max.
- Addressing modes
  - single and dual address modes (for DREQ activated devices)
  - address increment of -2, -1, 0, 1, 2
- Transfer request modes
  - external from DREQ
  - internally generated signals from A/D, ITU, SCI
- Cycle steal and burst mode options
- Allows transfers to happen with no CPU loading

DMAC block diagram

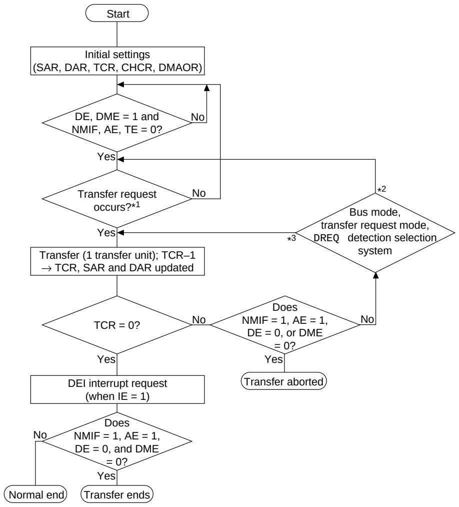
DMAOR: DMA operation register
SARn: DMA source address register
DARn: DMA destination address register
TCRn: DMA transfer count register
CHCRn: DMA channel control register
DEIn: DMA transfer-end interrupt request to CPU
n: 0–3



DMAC registers

Channel	Name	Abbreviation	R/W	Initial Value	Address	Access Size
0	DMA source address register 0	SAR0*3	R/W	Undefined	H'5FFFF40	16, 32
	DMA destination address register 0	DAR0*3	R/W	Undefined	H'5FFFF44	16, 32
	DMA transfer count register 0	TCR0*3	R/W	Undefined	H'5FFFF4A	16, 32
	DMA channel control register 0	CHCR0	R/(W)*1	H'0000	H'5FFFF4E	8, 16, 32
1	DMA source address register 1	SAR1*3	R/W	Undefined	H'5FFFF50	16, 32
	DMA destination address register 1	DAR1*3	R/W	Undefined	H'5FFFF54	16, 32
	DMA transfer count register 1	TCR1*3	R/W	Undefined	H'5FFFF5A	16, 32
	DMA channel control register 1	CHCR1	R/(W)*1	H'0000	H'5FFFF5E	8, 16, 32
2	DMA source address register 2	SAR2*3	R/W	Undefined	H'5FFFF60	16, 32
	DMA destination address register 2	DAR2*3	R/W	Undefined	H'5FFFF64	16, 32
	DMA transfer count register 2	TCR2*3	R/W	Undefined	H'5FFFF6A	16, 32
	DMA channel control register 2	CHCR2	R/(W)*1	H'0000	H'5FFFF6E	8, 16, 32
3	DMA source address register 3	SAR3*3	R/W	Undefined	H'5FFFF70	16, 32
	DMA destination address register 3	DAR3*3	R/W	Undefined	H'5FFFF74	16, 32
	DMA transfer count register 3	TCR3*3	R/W	Undefined	H'5FFFF7A	16, 32
	DMA channel control register 3	CHCR3	R/(W)*1	H'0000	H'5FFFF7E	8, 16, 32
Shar-ed	DMA operation register	DMAOR	R/(W)*2	H'0000	H'5FFFF48	8, 16, 32

DMAC Usage



Start

Initial settings (SAR, DAR, TCR, CHCR, DMAOR)

DE, DME = 1 and NMIF, AE, TE = 0?

No

Yes

Transfer request occurs?*1

No

Yes

Transfer (1 transfer unit); TCR–1 → TCR, SAR and DAR updated

TCR = 0?

No

Yes

DEI interrupt request (when IE = 1)

Does NMIF = 1, AE = 1, DE = 0, and DME = 0?

No

Yes

Normal end

Transfer ends

Does NMIF = 1, AE = 1, DE = 0, or DME = 0?

No

Yes

Transfer aborted

*2

Bus mode, transfer request mode, DREQ detection selection system

*3

Notes: 1. In auto-request mode, transfer begins when NMIF, AE, and TE are all 0 and the DE and DME bits are set to 1.

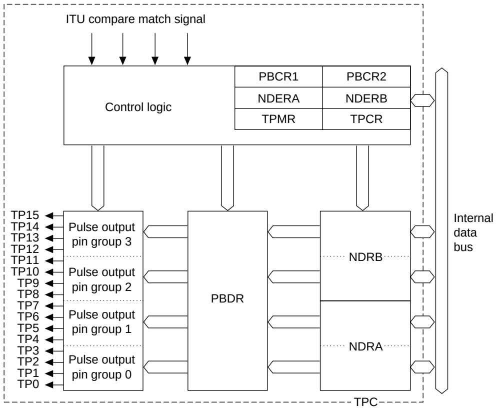
2. DREQ = level detection in burst mode (external request), or cycle steal mode.

3. DREQ = edge detection in burst mode (external request), or auto request mode in burst mode.

Timing Pattern Controller

- 4 bits x 4 TPC units digital output
- typically used in conjunction with ITU and DMAC
- used to synthesize pulse trains whose details are determined by ITU timing and a sequence of bits stored in memory
  - good for doing serial protocols

TPC Block Diagram



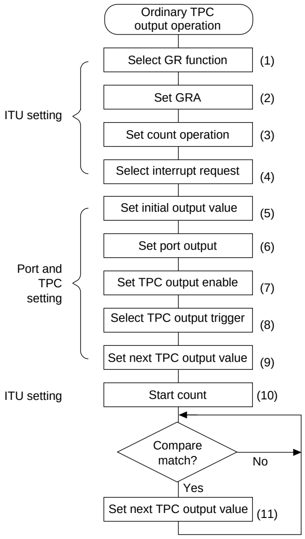
TPMR: TPC output mode register
TPCR: TPC output control register
NDERB: Next data enable register B
NDERA: Next data enable register A

PBCR1: Port B control register 1
PBCR2: Port B control register 2
NDRB: Next data register B
NDRA: Next data register A
PBDR: Port B data register

TPC Registers

Name	Abbreviation	R/W	Initial Value	Address*[1]	Access Size
Port B control register 1	PBCR1	R/W	H'0000	H'5FFFFCC	8, 16
Port B control register 2	PBCR2	R/W	H'0000	H'5FFFFCE	8, 16
Port B data register	PBDR	R/(W)*[2]	H'0000	H'5FFFFC2	8, 16
TPC output mode register	TPMR	R/W	H'F0	H'5FFFFF0	8, 16
TPC output control register	TPCR	R/W	H'FF	H'5FFFFF1	8, 16
Next data enable register B	NDERB	R/W	H'00	H'5FFFFF2	8, 16
Next data enable register A	NDERA	R/W	H'00	H'5FFFFF3	8, 16
Next data register A	NDRA	R/W	H'00	H'5FFFFF5/ H'5FFFFF7*[3]	8, 16
Next data register B	NDRB	R/W	H'00	H'5FFFFF4/ H'5FFFFF6*[3]	8, 16

TPC usage flow chart



Ordinary TPC output operation

ITU setting

- Select GR function (1)
- Set GRA (2)
- Set count operation (3)
- Select interrupt request (4)

Port and TPC setting

- Set initial output value (5)
- Set port output (6)
- Set TPC output enable (7)
- Select TPC output trigger (8)
- Set next TPC output value (9)

ITU setting

- Start count (10)

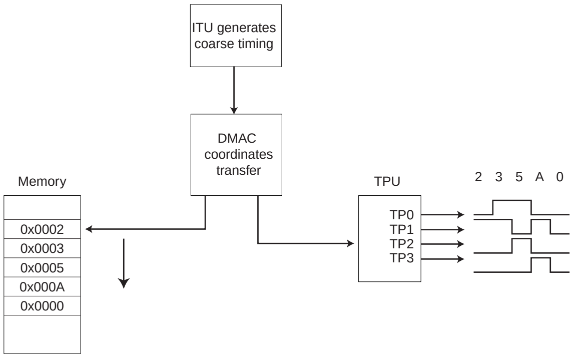
Compare match?

No

Yes

Set next TPC output value (11)

TPC + DMA + ITU example



ITU generates coarse timing

DMAC coordinates transfer

Memory

0x0002
0x0003
0x0005
0x000A
0x0000

TPU

TP0
TP1
TP2
TP3

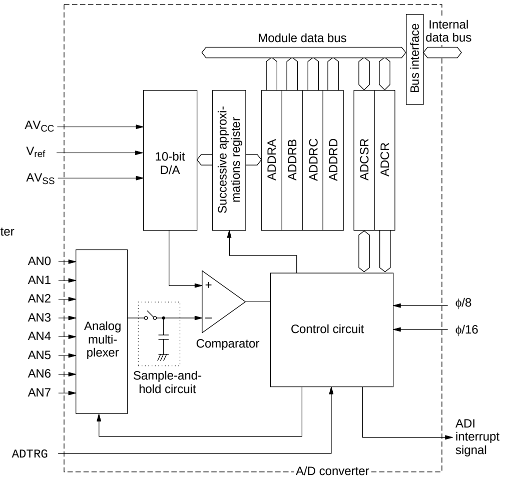
2 3 5 A 0

A/D Converter

- 8 channels, multiplexed
- 10 bits/channel
- Integrated S/H with internal or external sample trigger
- Successive Approximation
- Vref tied to 2.5V on SH1WH => 2.44 mV/LSB
- 11.2 us per channel conversion rate
- Can trigger DMAC for background sampling
- Scan mode capable (sample from each channel round-robin style)
- polled or interrupt mode for A/D completion

A/D Converter Block Diagram

ADCR: A/D control register
ADCSR: A/D control/status register
ADDRA: A/D data register A
ADDRB: A/D data register B
ADDRC: A/D data register C
ADDRD: A/D data register D



A/D converter registers

Register Name	Abbreviation	R/W	Initial Value	Address*[1]	Access Size
A/D data register A (high)	ADDRAH	R	H'00	H'05FFFEE0	8, 16
A/D data register A (low)	ADDRAL	R	H'00	H'05FFFEE1	16
A/D data register B (high)	ADDRBH	R	H'00	H'05FFFEE2	8, 16
A/D data register B (low)	ADDRBL	R	H'00	H'05FFFEE3	16
A/D data register C (high)	ADDRCH	R	H'00	H'05FFFEE4	8, 16
A/D data register C (low)	ADDRCL	R	H'00	H'05FFFEE5	16
A/D data register D (high)	ADDRDH	R	H'00	H'05FFFEE6	8, 16
A/D data register D (low)	ADDRDL	R	H'00	H'05FFFEE7	16
A/D control/status register	ADCSR	R/(W)*[2]	H'00	H'05FFFEE8	8, 16
A/D control register	ADCR	R/W	H'7F	H'05FFFEE9	8, 16

Embedded RAM

- 8K of fast RAM
  - 1 cycle access time read/write
  - 32-bit wide datapath
- Use for inner loops and time-critical processing
  - like a user managed cache
- Located from 0xF000000 to 0xF002000

Power Down Modes

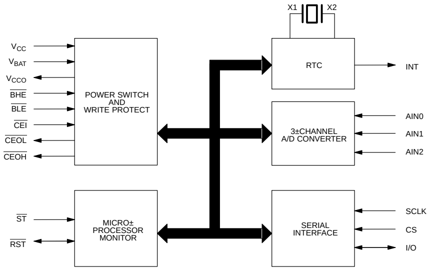
- Power down modes entered by executing SLEEP instruction
  - sleep or standby set by changing a bit in the SBYCR
- sleep mode
  - clock runs, CPU halts, all peripherals run
  - exit through interrupt, DMA address error, power on reset, manual reset
- standby mode
  - clock halts, all functions stop
  - exit through NMI, power on reset, or manual reset

Building a system

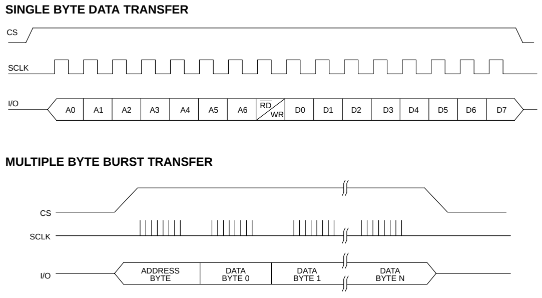
- Minimal SH-1 system
  - SH-1 + embedded ROM/RAM, crystal
- SH1WH system
  - SH-1
  - 8 MB DRAM
  - 128K battery backed SRAM
  - 1 MB ISP FLASH
  - D/A converter
  - RTC
  - RS-232 tranceivers
  - Extra digital I/O
  - Switching regulator

SH1WH RTC

- Dallas Semiconductor DS1670 system controller/RTC
- Serial interface to RTC
  - CLK, DIN/DOUT, CS
- Power fail manager
  - turns of SRAM CS, resets processor on low Vcc condition
- 3-channel, 8-bit A/D converter
  - used for temperature sensor
- Wake-up alarm clock
- Watchdog timer



RTC registers, serial interface

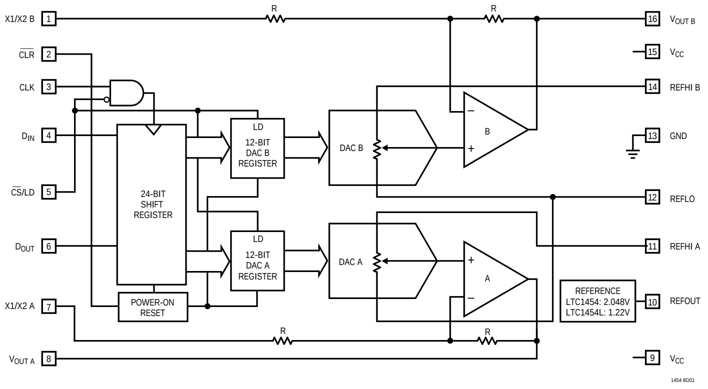


	BIT7							BIT0
00	0	10 SECONDS			SECONDS			
01	0	10 MINUTES			MINUTES			
02	0	12/24	10 HR A/P	10 HR	HOURS			
03	0	0	0	0	0	DAY		
04	0	0	10 DATE		DATE			
05	0	0	0	10 MO.	MONTH			
06	10 YEAR				YEAR			
07	M	10 SEC ALARM			SECONDS ALARM			
08	M	10 MIN ALARM			MINUTES ALARM			
09	M	12/24	10 HR A/P	10 HR	HOUR ALARM			
0A	M	0	0	0	DAY ALARM			
0B	CONTROL REGISTER							
0C	STATUS REGISTER							
0D	WATCHDOG REGISTER							
0E	ADC REGISTER							
0F 7F	RESERVED							

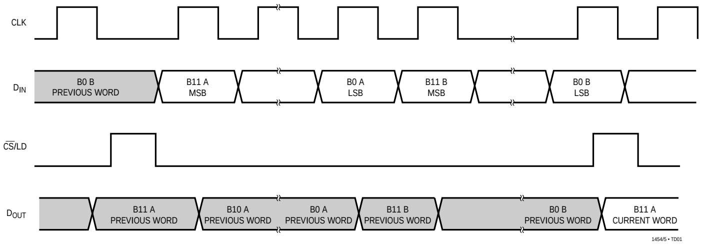
D/A converter

- LTC1454L D/A converter
  - 12-bit resolution
  - 1.220V reference + x2 amplifier => 2.44V effective full-scale
  - power-on reset to 0
  - 14 us typ. settling time, 1 V/us typ. slew rate

D/A converter



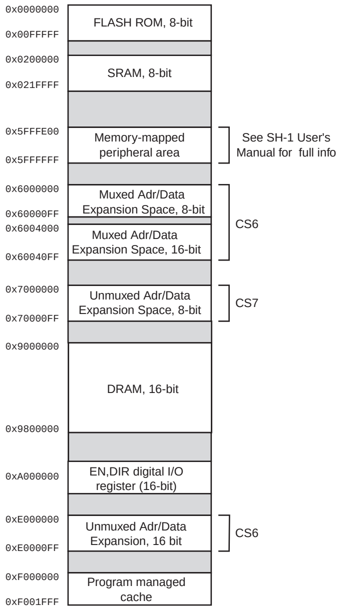
D/A converter



FLASH ROM

- 1 MB, 3.3V only boot-block ROM with autoshutdown
- embedded algorithms for erase and program
  - writing a special sequence to the FLASH causes erasing and programming to occur
- like all FLASH, an entire sector must be erased before being written to
  - erase defaults to ‘1’
  - programming will clear respective bits to ‘0’
  - possible to program over if all you’re doing is setting bits to 0
- drivers provided for your convenience in using the FLASH
  - not recommended to muck around too much with the boot block, as losing the boot block means you’ll have to desolder the ROM and burn it in a programmer

SH1WH Memory Map



0x0000000
FLASH ROM, 8-bit
0x00FFFFF

0x0200000
SRAM, 8-bit
0x021FFFF

0x5FFFE00
Memory-mapped peripheral area
0x5FFFFFF
See SH-1 User's Manual for full info

0x6000000
Muxed Adr/Data Expansion Space, 8-bit
0x60000FF
0x6004000
Muxed Adr/Data Expansion Space, 16-bit
0x60040FF
CS6

0x7000000
Unmuxed Adr/Data Expansion Space, 8-bit
0x70000FF
CS7

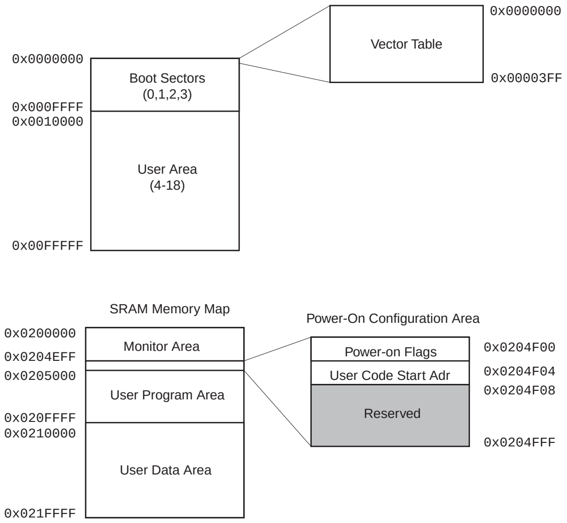
0x9000000
DRAM, 16-bit
0x9800000

0xA000000
EN,DIR digital I/O register (16-bit)

0xE000000
Unmuxed Adr/Data Expansion, 16 bit
0xE0000FF
CS6

0xF000000
Program managed cache
0xF001FFF

SRAM and FLASH Memory Map



0x0000000
Boot Sectors (0,1,2,3)
0x000FFFF
0x0010000
User Area (4-18)
0x00FFFFF

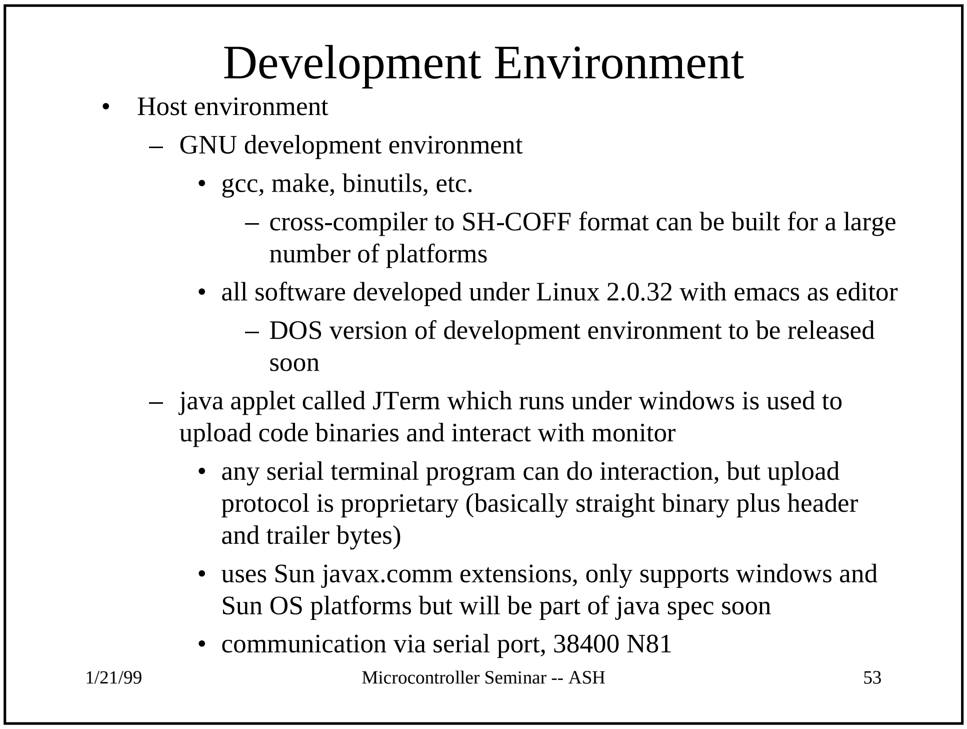
0x0000000
Vector Table
0x00003FF

SRAM Memory Map

0x0200000
Monitor Area
0x0204EFF
0x0205000
User Program Area
0x020FFFF
0x0210000
User Data Area
0x021FFFF

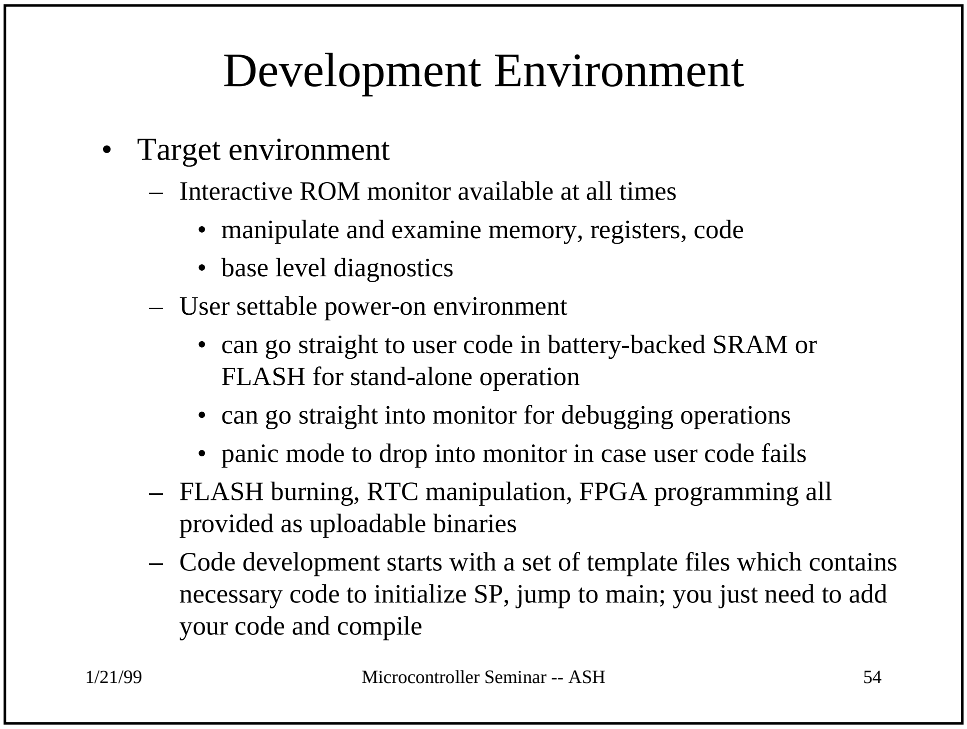
Power-On Configuration Area

Power-on Flags
0x0204F00
User Code Start Adr
0x0204F04
0x0204F08
Reserved
0x0204FFF



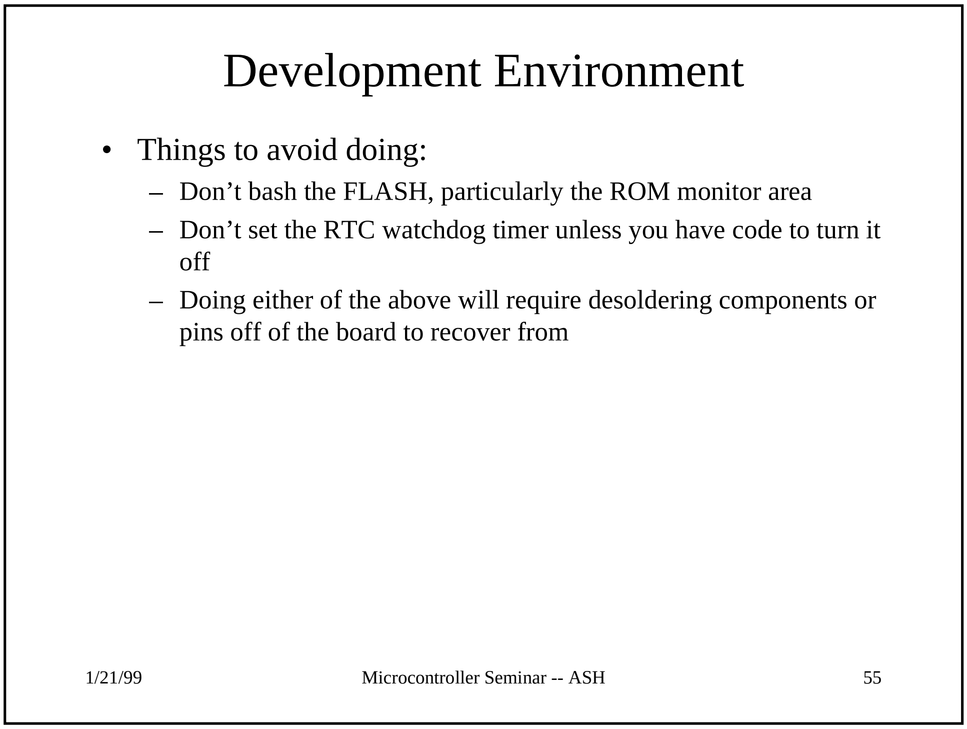
Development Environment

- Host environment
  - GNU development environment
    - gcc, make, binutils, etc.
      - cross-compiler to SH-COFF format can be built for a large number of platforms
    - all software developed under Linux 2.0.32 with emacs as editor
      - DOS version of development environment to be released soon
  - java applet called JTerm which runs under windows is used to upload code binaries and interact with monitor
    - any serial terminal program can do interaction, but upload protocol is proprietary (basically straight binary plus header and trailer bytes)
    - uses Sun javax.comm extensions, only supports windows and Sun OS platforms but will be part of java spec soon
    - communication via serial port, 38400 N81



Development Environment

- Target environment
  - Interactive ROM monitor available at all times
    - manipulate and examine memory, registers, code
    - base level diagnostics
  - User settable power-on environment
    - can go straight to user code in battery-backed SRAM or FLASH for stand-alone operation
    - can go straight into monitor for debugging operations
    - panic mode to drop into monitor in case user code fails
  - FLASH burning, RTC manipulation, FPGA programming all provided as uploadable binaries
  - Code development starts with a set of template files which contains necessary code to initialize SP, jump to main; you just need to add your code and compile



Development Environment

- Things to avoid doing:
  - Don’t bash the FLASH, particularly the ROM monitor area
  - Don’t set the RTC watchdog timer unless you have code to turn it off
  - Doing either of the above will require desoldering components or pins off of the board to recover from