R80515 User Manual

- 8 Bit CISC CPU (Rev1.2)

CONTENTS

1 Introduction

The R80515 is a core of a fast single-chip 8-bit microcontroller. It is a fully functional 8-bit embedded that executes all ASM51 instructions.

1.1 Features

- ➢ Control Unit
- ■ 8-bit Instruction decoder
- ■ Reduced instruction cycle time up to 12 times
- ➢ Arithmetic-Logic Unit
- ■ 8-bit arithmetic and logical operations
- ■ Boolean manipulations
- ■ 8x8 bit multiplication and 8/8 bit division
- ➢ Multiplication-Division Unit
- ■ 16x16 bit multiplication
- ■ 32/16 bit and 16/16 bit division
- ■ 32 bit normalization
- ■ 32 bit L/R shifting
- ■ 32-bit Input/Output ports
- ➢ Four 8-bit I/O ports
- ■ Alternate port functions such as external interrupts and serial interface are separated, providing extra port pins when compared with standard 8051
- ➢ Three 16-bit Timer/Counters
- ■ Compare/Capture Unit
- ■ Four 16-bit Compare registers used for Pulse Width Modulation
- ■ Four external Capture inputs used for Pulse Width Measuring
- ■ 16-bit Reload register used for Pulse Generation.
- ➢ Two Serial Peripheral Interfaces in full duplex mode
- ■ Synchronous mode, fixe baud rate, Serial 0 only
- ■ 8-bit UART mode, variable baud rate
- ■ 9-bit UART mode, variable baud rate
- ■ Additional Baud Rate Generator for Serial 0
- ➢ Interrupt Controller
- ■ Four priority levels with 13 interrupt sources
- ➢ 15 bit Programmable Watchdog Timer
- ➢ Internal Data Memory interface
- ■ Can address up to 256B of Data Memory Space
- ➢ External Memory interface
- ■ Has independent Data and Program Memory interface
- ■ Can address up to 64 KB of External Program Memory
- ■ Can address up to 64 KB of External Data Memory
- ➢ Special Function Registers interface
- ■ Services up to 72 External Special Function Registers

- Power Management Unit
- Power management modes IDLE and STOP

2 Block Diagram

2.1 Block Diagram

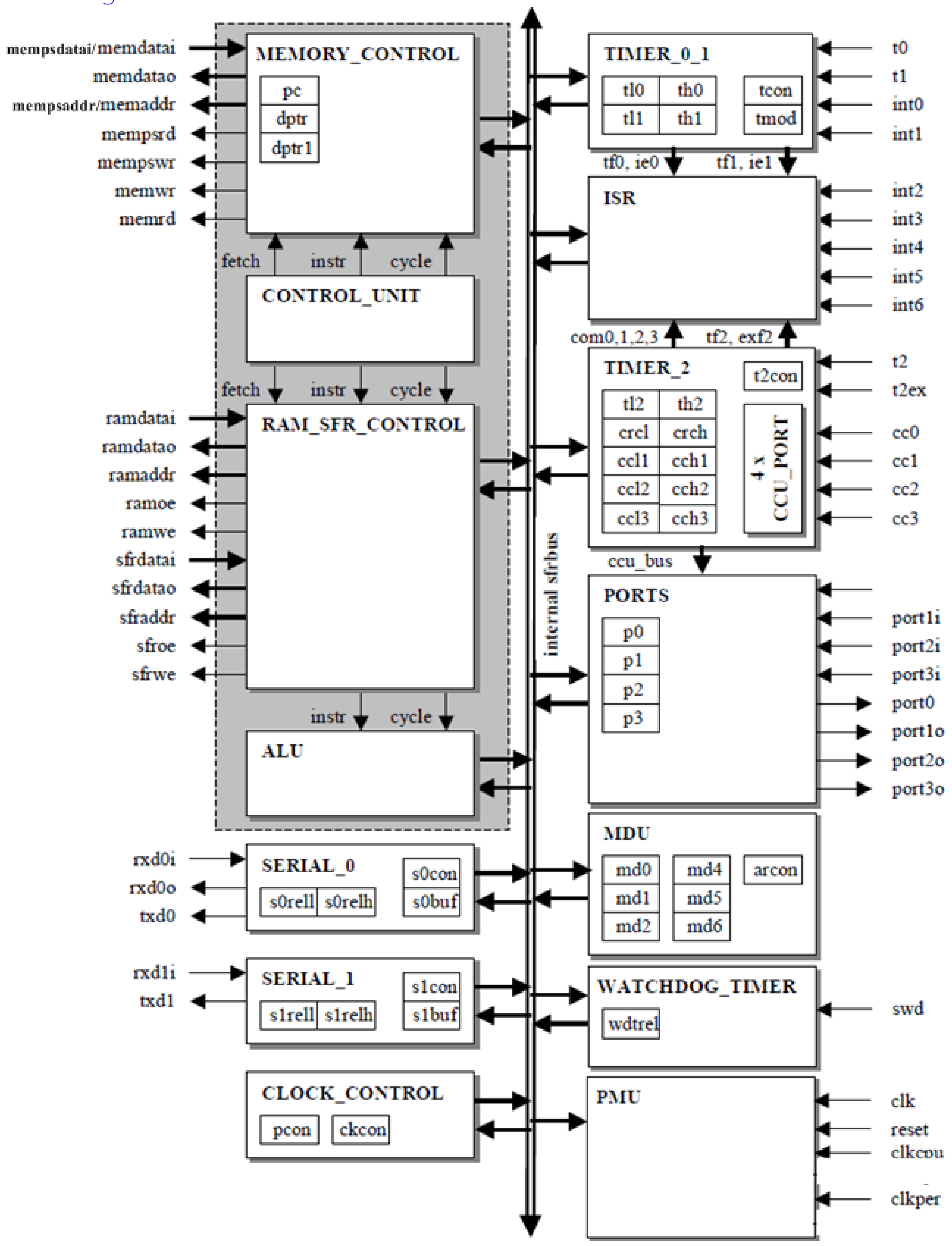


Figure 2-1 Block Diagram

3 Interface

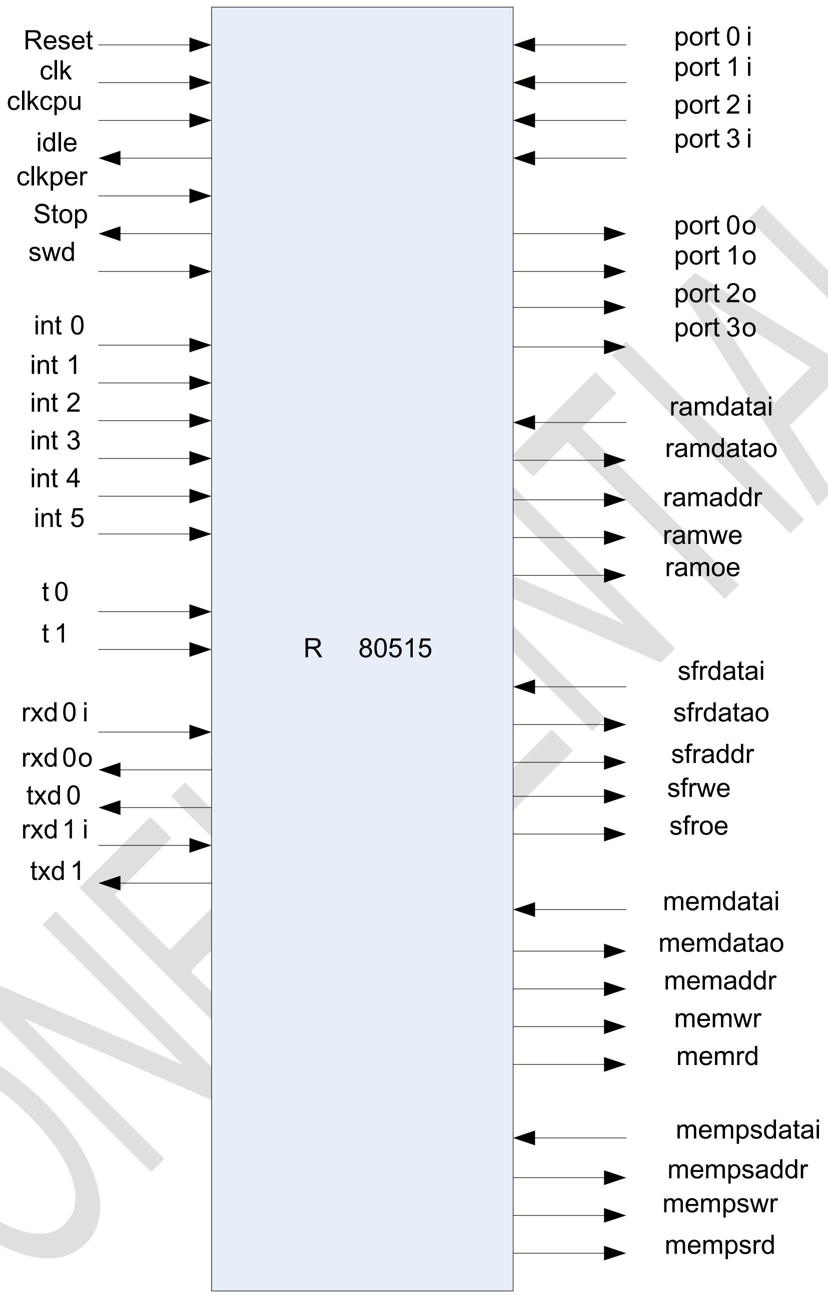


Figure 3-1 R80515 interface diagram

3.1 Interface Signal

Name	Width	I/O	Active State	Registered	Description
System Control Interface					
Clk	1	I	N/A	N/A	Global clock
Clkcpu	1	I	N/A	N/A	CPU clock input
idle	1	O	N/A	No	idle status
Clkper	1	I	N/A	N/A	Peripheral clock input
stop	1	O	N/A	No	stop status
Reset	1	I	High	N/A	Hardware reset
Swd	1	I	High		Start Watchdog Timer
Port					
Port0i	8	I	N/A		P0 input
Port1i	8	I	N/A		P1 input
Port2i	8	I	N/A		P2 input
Port3i	8	I	N/A		P3 input
Port0o	8	O	N/A		P0 output
Port1o	8	O	N/A		P1 output
Port2o	8	O	N/A		P2 output
Port3o	8	O	N/A		P3 output
External Interrupt/Port alternate Interface					
Int0	1	I	Low/ fall		External interrupt 0
Int1	1	I	Low/ fall		External interrupt 1
Int2	1	I	fall/rise		External interrupt 2
Int3	1	I	fall/rise		External interrupt 3
Int4	1	I	rise		External interrupt 4
Int5	1	I	rise		External interrupt 5
Int6	1	I	rise		External interrupt 6
Serial/Port alternate Interface					
Rxd0i	1	I	N/A	N/A	Serial 0 receive data
Rxd1i	1	I	N/A	N/A	Serial 1 receive data
Rxd0o	1	O	N/A	Yes	Serial 0 receive clock
Txd0	1	O	N/A	Yes	Serial 0 transmit data
Txd1	1	O	N/A	Yes	Serial 1 transmit data
Timer/Port alternate Interface					
T0	1	I	fall	N/A	Timer0 external input
T1	1	I	fall	N/A	Timer1 external input
External Memory Interface					

Memdatai	8	I	N/A	N/A	External Data Memory read data input
Memdatao	8	O	N/A	No	External Data Memory Write data output
Memaddr	16	O	N/A	No	External Data Memory Address bus
Memwr	1	O	High	Yes	External Data Memory write enable
Memrd	1	O	High	Yes	External Data Memory read enable
mempsdatai	8	I	N/A	N/A	Program Memory read data input
Mempsaddr	16	O	N/A	Yes	Program Memory Address bus
Mempsrd	1	O	High	Yes	Program Memory read enable
Mempswr	1	O	High	Yes	Program Memory write enable
Internal Data Memory Interface					
Ramdatai	8	I	N/A	N/A	Internal Data Memory Data input
Ramdatao	8	O	N/A	Yes	Internal Data Memory Data output
Ramaddr	8	O	N/A	Yes	Internal Data Memory address bus
Ramwe	1	O	High	Yes	Internal Data Memory Write enable
Ramoe	1	O	High	Yes	Internal Data Memory Read enable
Sfrdatai	8	I	N/A	N/A	SFR data input
Sfrdatao	8	O	N/A	Yes	SFR data output
Sfraddr	7	O	N/A	Yes	SFR address bus
Sfrwe	1	O	High	Yes	SFR write enable
Sfroe	1	O	High	Yes	SFR read enable

4 Register Description

4.1 Register Table

Name	Address Offset	Width(bit)	R/W	Default	Description
P0	00H	8	R/W	8'hFF	Port0
Sp	01H	8	R/W	8'h07	Stack Pointer
Dpl	02H	8	R/W	8'h00	Data Pointer Low0
Dph	03H	8	R/W	8'h00	Data Pointer High0
Dpl1	04H	8	R/W	8'h00	Data Pointer Low1
Dph1	05H	8	R/W	8'h00	Data Pointer High1
Wdtrel	06H	8	R/W	8'h00	Watchdog Timer Reload
Pcon	07H	8	R/W	8'h00	Power Control
Tcon	08H	8	R/W	8'h00	Timer/Counter Control
Tmod	09H	8	R/W	8'h00	Timer Mode Control
Tl0	0AH	8	R/W	8'h00	Timer0, low byte
Tl1	0BH	8	R/W	8'h00	Timer1, low byte
Th0	0CH	8	R/W	8'h00	Timer0, high byte
Th1	0DH	8	R/W	8'h00	Timer1, high byte
Ckcon	0EH	8	R/W	8'h01	Clock Control(stretch=1)
–	0F	8			
P1	10H	8	R/W	8'hFF	Port1
–	11H	8	–	–	–
Dps	12H	8	R/W	8'h00	Data Pointer Select
–	13H	8	–	–	–
–	14H	8	–	–	–
–	15H	8	–	–	–
–	16H	8	–	–	–
–	17H	8	–	–	–
S0con	18H	8	R/W	8'h00	Serial Port0 Control
S0buf	19H	8	R/W	8'h00	Serial Port0, Data Buffer
Ien2	1AH	8	R/W	8'h00	Interrupt Enable2
S1con	1BH	8	R/W	8'h00	Serial Port1 Control
S1buf	1CH	8	R/W	8'h00	Serial Port1, Data Buffer
S1rell	1DH	8	R/W	8'h00	Serial Port1, Reload, low byte
–	1EH	8	–	–	–
–	1FH	8	–	–	–
P2	20H	8	R/W	8'h00	Port2
–	21H	8	—	–	–
–	22H	8	—	–	–
–	23H	8	—	–	–
–	24H	8	—	–	–

–	25H	8	—	–	–
–	26H	8	—	–	–
–	27H	8	—	–	–
Ien0	28H	8	R/W	8'h00	Interrupt Enable0
Ip0	29H	8	R/W	8'h00	Interrupt priority 0
S0rell	2AH	8	R/W	8'hD9	Serial Port 0, Reload, low byte
–	2BH	8	—	–	–
–	2CH	8	—	–	–
–	2DH	8	—	–	–
–	2EH	8	—	–	–
–	2FH	8	—	–	–
P3	30H	8	R/W	8'hFF	Port3
–	31H	8	—	–	–
–	32H	8	—	–	–
–	33H	8	—	–	–
–	34H	8	—	–	–
–	35H	8	—	–	–
–	36H	8	—	–	–
–	37H	8	—	–	–
Ien1	38H	8	R/W	8'h00	Interrupt Enable 1
Ip1	39H	8	R/W	8'h00	Interrupt Priority 1
S0relh	3AH	8	R/W	8'h03	Serial Port 0, Reload, high byte
S1relh	3BH	8	R/W	8'h00	Serial Port 1, Reload, high byte
–	3CH	8	—	–	–
–	3DH	8	—	–	–
–	3EH	8	—	–	–
–	3FH	8	—	–	–
Ircon	40H	8	R/W	8'h00	Interrupt Request Control
–	41H	8	–	8'h00	–
–	42H	8	–	8'h00	–
–	43H	8	–	8'h00	–
–	44H	8	–	8'h00	–
–	45H	8	–	8'h00	–
–	46H	8	–	8'h00	–
–	47H	8	–	8'h00	–
T2con	48H	8	R/W	8'h00	Timer2 Control only bit[6:5] are used for SD project

–	49H	8	–	–	–
–	4AH	8	–	8'h00	–
–	4BH	8	–	8'h00	–
–	4CH	8	–	8'h00	–
–	4DH	8	–	8'h00	–
–	4EH	8	–	–	–
–	4FH	8	–	–	–
psw	50H	8	R/W	8'h00	Program Status Word
–	51H	8	–	–	–
–	52H	8	–	–	–
–	53H	8	–	–	–
–	54H	8	–	–	–
–	55H	8	–	–	–
–	56H	8	–	–	–
–	57H	8	–	–	–
Wdcon	58H	8	R/W	8'h00	Baud rate generate Control (only bit7 is used)
–	59H	8	–	–	–
–	5AH	8	–	–	–
–	5BH	8	–	–	–
–	5CH	8	–	–	–
–	5DH	8	–	–	–
–	5EH	8	–	–	–
–	5FH	8	–	–	–
Acc	60H	8	R/W	8'h00	Accumulator
–	61H	8	–	–	–
–	62H	8	–	–	–
–	63H	8	–	–	–
–	64H	8	–	–	–
–	65H	8	–	–	–
–	66H	8	–	–	–
–	67H	8	–	–	–
–	68H	8	–	–	–
Md0	69H	8	R/W	8'h00	Multiplication/Division Register0
Md1	6AH	8	R/W	8'h00	Multiplication/Division Register1
Md2	6BH	8	R/W	8'h00	Multiplication/Division Register2
Md3	6CH	8	R/W	8'h00	Multiplication/Division Register3

Md4	6DH	8	R/W	8’h00	Multiplication/Division Register4
Md5	6EH	8	R/W	8’h00	Multiplication/Division Register5
Arcon	6FH	8	R/W	8’h00	Arithmetic Control
B	70H	8	R/W	8’h00	B register
–	71H	8	–	–	–
–	72H	8	–	–	–
–	73H	8	–	–	–
–	74H	8	–	–	–
–	75H	8	–	–	–
–	76H	8	–	–	–
–	77H	8	–	–	–
–	78H	8	–	–	–
–	79H	8	–	–	–
–	7AH	8	–	–	–
	7BH	8	–	–	–not used
	7CH	8	–	–	–not used
–	7DH	8	–	–	–
–	7EH	8	–	–	–
Resevered	7FH	8	R/W	8’h00	Not used

4.2 Detailed Register Description

4.2.1 P0/P1/P2/P3 (80H/90H/A0H/B0H)

The contents of the SFR P0-P3 can be observed on corresponding pins on the chip. Writing a ‘1’ to any of the ports cause the corresponding pin to be held at high level (VCC), and writing a ‘0’ causes the corresponding pin to be held at low level (GND).

All four ports on the chip are bi-directional. Each of them consists of a Latch (SFR P0-P3), an output driver, and an input buffer, so the MCU can output or read data through any of the ports if they are not used for alternate purposes.

4.2.2 Sp (81H)

The stack pointer (SP) is a 1-byte register initialized to 07H after reset. This register is incremented before PUSH and LCALL/ACALL instructions, causing the stack to begin at location 08H.

4.2.3 Dpl/dph/dp1l/dp1h (82H/83H/84H/85H)

The data pointer (dptr) is 2 bytes wide. The lower part is dpl, and the highest is dph. It is generally used to access external code or data space (through MOVC A, @A+DPTR, MOVX A, @DPTR, MOVX @DPTR, A).

There is Dual Data Pointer in the R80515. The standard DPTR is a 16-bit register that is used to address external memory or peripherals. In the R80515 core the standard data pointer is called DPTR (dpl/dph), the second data pointer is called DPTR1 (dp1l/dp1h). The data pointer select bit (dps.0) chooses the active pointer. All DPTR-related instructions use the currently selected DPTR for any activity.

4.2.4 Wdtrel (86H)

Register addr	86H	
Bit number	7	6:0

Bit field name	wdtrels	wdtrel
R/W	R/W	R/W
default	1'b0	7'b0

Bits	Name	R/W	Description
7	wdtrels	R/W	Prescaler select bit. If set, the watchdog is clocked through an additional divide-by-16 prescaler.
6:0	wdtrel	R/W	7-bit reload value for the high-byte of the watchdog timer. This value is loaded to the wdt when a refresh is triggered by a consecutive setting of the bits wdt (ien0.6) and swdt (ien1.6).

4.2.5 Pcon (87H)

Register addr	87H					
Bit number	7	6:4	3	2	1	0
Bit field name	Smod	-	gf1	gf0	stop	idle
R/W	R/W	RO	R/W	R/W	R/W	R/W
default	1'b0	3'b0	1'b0	1'b0	1'b0	1'b0

Bits	Name	R/W	Description
7	smod	R/W	Used to define the baud rate for serial 0 mode 1,2 and 3.
6:4	Reserved	RO	Reserved
3	gf1	R/W	
2	gf0	R/W	
1	stop	WO	read as 0; write 1 system entry stop mode
0	idle	WO	read as 0; write 1 system entry idle mode

4.2.6 Tcon (88H)

Register addr	88H							
Bit number	7	6	5	4	3	2	1	0
Bit field name	Tf1	Tr1	Tf0	Tr0	Ie1	It1	Ie0	It0
R/W	R/W/HC	R/W	R/W/HC	R/W	R	R/W	R	R/W
default	1'b0	1'b0	1'b0	1'b0	1'b0	1'b0	1'b0	1'b0

Bits	Name	R/W	Description
7	Tf1	R/W/HC	Timer1 overflow flag set by hardware when Timer1 overflows. This flag can be cleared by software and is automatically cleared when interrupt is processed Timer1 overflow conditions: | | conditions |

			{t1.m1: t1.m0}=2’b00 {t1.m1: t1.m0}=2’b01 — thl overflow {t1.m1: t1.m0}=2’b10 — tl1 overflow {t0.m1: t0.m0}=2’b11 — th0 overflow
6	Tr1	R/W	Timer1 Run control bit. If cleared, Timer1 stops
5	Tf0	R/W/HC	Timer0 overflow flag set by hardware when Timer0 overflows. This flag can be cleared by software and is automatically cleared when interrupt is processed Timer0 overflow conditions: conditions {t0.m1: t0.m0}=2’b00 {t0.m1: t0.m0}=2’b01 — th0 overflow {t0.m1: t0.m0}=2’b10 {t0.m1: t0.m0}=2’b11 — tl0 overflow
4	Tr0	R/W	Timer0 Run control bit. If cleared, Timer0 stops
3	Ie1	R	Interrupt1 edge flag. Set by hardware, when falling edge on external pin int1 is observed. Cleared when interrupt is processed If it1=0 (low level), hardware will change this bit based on pin int1: Pin int1=0, this bit will be set Pin int1=1, this bit will be reset If it1=1(falling edge), hardware will set this bit after detect the falling edge of pin int1
2	It1	R/W	Interrupt1 type control bit. Selects falling edge or low level on input pin to cause interrupt 0: low level 1: falling edge
1	Ie0	R	Interrupt0 edge flag. Set by hardware, when falling edge on external pin int0 is observed. Cleared when interrupt is processed If it0=0 (low level), hardware will change this bit based on pin int0: Pin int0=0, this bit will be set Pin int0=1, this bit will be reset If it0=1(falling edge), hardware will set this bit after detect the falling edge of pin int0
0	It0	R/W	Interrupt0 type control bit. Selects falling edge or low level on input pin to cause interrupt 0: low level 1: falling edge

4.2.7 Tmod (89H)

Register addr	89H							
Bit number	7	6	5	4	3	2	1	0
Bit field name	Gate (timer1)	c/t (timer1)	M1 (timer1)	M0 (timer1)	Gate (timer0)	c/t (timer0)	M1 (timer0)	M0 (timer0)
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
default	1'b0	1'b0	1'b0	1'b0	1'b0	1'b0	1'b0	1'b0

Bits	Name	R/W	Description
7/3	Gate	R/W	External gate control enable/disable. 0: enable external gate control. If tr0/tr1 in tcon set, and this bit is 0, the external gate is enabled. And if detect the falling edge of int0 or int1, the external gate is enabled too. After external gate control enable, if detect int0 or int1 falling edge, and current operation is counter, then a counter will increase every falling edge of t0 or t1. 1: disable external gate control.
6/2	c/t	R/W	Timer or counter operation selects. 1: counter 0: timer
5/1	M1	R/W	Selects mode for Timer/Counter 0 or Timer/Counter 1.
4/0	M0	R/W	Selects mode for Timer/Counter 0 or Timer/Counter 1.

M1	M0	Mode	Function
0	0	Mode0	13-bit counter/timer. 5 lower bits in tl0 or tl1 and 8 bits in th0 or th1. The 3 high order bits of tl0 or tl1 are hold zeros.
0	1	Mode1	16-bit counter/timer
1	0	Mode2	8-bit auto-reload counter/timer. The reload value is kept in th0 or th1 which is incremented every cycle. When tl0/tl1 overflows, a value from th0/th1 is copied to t10/tl1
1	1	Mode3	Timer1: stop Timer0: act as two independent 8 bit timer/counter

4.2.8 Tl0/tl1/th0/th1 (8AH/8BH/8CH/8DH)

Timer0 and Timer1 register.

Mode	Function
Mode0	13-bit timer0/timer1 5 lower bits in lower tl0 or tl1, and 8 higher bits in th0 or th1. The higher 3 bits of tl0 and tl1 are hold zero
Mode1	16-bit timer0/timer1 The lower 8-bits in tl0 or tl1, and the higher 8-bits in th0 or th1

Mode2	8-bit auto-reload timer0/timer1. The reload value is kept in th0 or th1, while tl0 or tl1 is incremented every machine cycle. When tl0/tl1 overflows, a value from th0/th1 is copied to tl0/tl1
Mode3	tl1 and th1 are not valid tl0 and th0 act as two independent 8-bit timer

Bits	Name	R/W	Description
7	Reserved	R	
6:4		R	
3	Reserved	R	
2:0			

4.2.9 S0con (98H)

Register addr	98H							
Bit number	7	6	5	4	3	2	1	0
Bit field name	sm0	sm1	sm20	ren0	tb80	rb80	ti0	ri0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
default	1’b0	1’b0	1’b0	1’b0	1’b0	1’b0	1’b0	1’b0

Bits	Name	R/W	Description
7	sm0	R/W	Sets baud rate
6	sm1	R/W	Sets baud rate

sm0	sm1	Mode	Description	Baud Rate
0	0	0	shift register	Fosc/12
0	1	1	8-bit UART	Variable
1	0	2	9-bit UART	Fclk/32 or 64 smod(pcon.7)=0: Fclk/64 smod=1: Fclk/32
1	1	3	9-bit UART	Variable

Bits	Name	R/W	Description
5	sm20	R/W	Serial 0 Multiprocessor communication function control 1: enable 0: disable
4	ren0	R/W	1: Enable serial 0 reception 0: Disable serial 0 reception
3	tb80	RO	The 9th transmitted data bit in Mode 2 and 3. CPU (master) will set or clear this bit.
2	rb80	R/W/SWC	The 9th received data bit in Mode 2 and 3. In mode1, if sm20=0, this bit is the stop bit.

			In mode0, this bit is not used Must be cleared by software.
1	ti0	R/W	Transmit completed interrupt flag. Hardware set after completion of a serial 0 transfer. Must be cleared by software.
0	ri0	R/W	Receive completed interrupt flag. Hardware set after completion of a serial 0 reception. Must be cleared by software.

4.2.10 S0buf/s1buf (99H/9CH)

Writing data to the SFR s0buf or s1buf will starts the transmission with different mode. Reading data from s0buf or s1buf for receive operation.

4.2.11 Ien2 (9AH) （NOT used in MCU Now）

Register addr	9AH							
Bit number	7	6	5	4	3	2	1	0
Bit field name	-	-	-	-	-	-	-	es1
R/W	RO	RO	RO	RO	RO	RO	RO	R/W
default	1'b0	1'b0	1'b0	1'b0	1'b0	1'b0	1'b0	1'b0

Bits	Name	R/W	Description
7:1	-	RO	Reserved
0	es1	R/W	0: disable serial channel 1 interrupt 1: enable

4.2.12 S1con (9BH)

Register addr	9BH							
Bit number	7	6	5	4	3	2	1	0
Bit field name	sm	-	sm21	ren1	tb81	rb81	ti1	ri1
R/W	R/W	RO	R/W	R/W	R/W	R/W	R/W	R/W
default	1'b0	1'b0	1'b0	1'b0	1'b0	1'b0	1'b0	1'b0

Bits	Name	R/W	Description
7	sm	R/W	Sets baud rate
6	-	RO	Reserved
5	sm21	R/W	Serial 1 Multiprocessor communication function control 1: enable 0: disable

4	ren1	R/W	1: Enable serial1 reception 0: Disable serial 1 reception
3	tb81	RO	The 9th transmitted data bit in Mode A. CPU (master) set or clear this bit.
2	rb81	R/W/SWC	The 9th received data bit in Mode 2 and 3. In modeB, if sm21=0, this bit is the stop bit. In mode0, this bit is not used Must be cleared by software.
1	ti1	R/W	Transmit completed interrupt flag. Hardware set after completion of a serial 1 transfer. Must be cleared by software.
0	ri1	R/W	Receive completed interrupt flag. Hardware set after completion of a serial 1 reception. Must be cleared by software.

4.2.13 S0rell/s0relh/s1rell/s1relh (AAH/BAH/9DH/BBH)

The serial channel 0 or serial channel1 reload register for computing baud rate.

4.2.14 Ien0 (A8H)

Register addr	A8H							
Bit number	7	6	5	4	3	2	1	0
Bit field name	eal	wdt	–	es0	et1	ex1	et0	ex0
R/W	R/W	R/W	RO	R/W	R/W	R/W	R/W	R/W
default	1'b0	1'b0	1'b0	1'b0	1'b0	1'b0	1'b0	1'b0

Bits	Name	R/W	Description
7	eal	R/W	0: disable all interrupts 1: enable all interrupts. Each interrupt has its own enable
6	wdt	R/W/HC	Watchdog timer refresh flag Set to initiate a refresh of the watchdog timer. Must be set directly before swdt (ien1.6) is set to prevent an unintentional refresh of the watchdog timer. This bit is reset by hardware 12 clock cycles after it has been set.
5	–	RO	Reserved
4	es0	R/W	0: disable serial 0 interrupt, includes transfer and receive 1: enable
3	et1	R/W	0: disable timer1 overflow interrupt 1: enable
2	ex1	R/W	0: disable external interrupt 1 1: enable
1	et0	R/W	0: disable timer 0 overflow interrupt 1: enable

0	ex0	R/W	0: disable external interrupt 0 1: enable

4.2.15 Ip0 (A9H)

Register addr	A9H							
Bit number	7	6	5	4	3	2	1	0
Bit field name	owds	wdts	ip0.5	ip0.4	ip0.3	ip0.2	ip0.1	ip0.0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
default	1'b0	1'b0	1'b0	1'b0	1'b0	1'b0	1'b0	1'b0

Bits	Name	R/W	Description
7	owds	R/W	Not used
6	wdts	RO	Watchdog timer status flag. Set by hardware when the watchdog timer was started. Can be read by software.
5:0	ip0.5-ip0.0	R/W	Group5-0 priority level control. The value of ip0.5-ip0.0 and ip1.5-ip1.0 define the priority level for interrupt group5-group0. ip1.x / ip0.x / priority level: 0 / 0 / level0 (lowest) 0 / 1 / level1 1 / 0 / level2 1 / 1 / level3 (highest)

4.2.16 Ien1 (B8H)

Register addr	B8H							
Bit number	7	6	5	4	3	2	1	0
Bit field name	–	swdt	ex6	ex5	ex4	ex3	ex2	eadc
R/W	RO	R/W	R/W	R/W	R/W	R/W	R/W	R/W
default	1'b0	1'b0	1'b0	1'b0	1'b0	1'b0	1'b0	1'b0

Bits	Name	R/W	Description
7	–	RO	Reserved
6	swdt	R/W/HC	Watchdog timer start/refresh flag Set to active/refresh the watchdog timer. If wdt (ien0.6) and swdt both set, a watchdog timer refresh is performed. This bit can be set to start watchdog timer and cleared to stop watchdog timer.
5	ex6	R/W	0: disable external interrupt 6 1: enable
4	ex5	R/W	0: disable external interrupt 5 1: enable

3	ex4	R/W	0: disable external interrupt 4 1: enable
2	ex3	R/W	0: disable external interrupt 3 1: enable
1	ex2	R/W	0: disable external interrupt 2 1: enable
0	eadc	R/W	0: disable A/D converter interrupt 1: enable

4.2.17 Ip1 (B9H)

Register addr	B9H							
Bit number	7	6	5	4	3	2	1	0
Bit field name	-	-	ip1.5	ip1.4	ip1.3	ip1.2	ip1.1	ip1.0
R/W	RO	RO	R/W	R/W	R/W	R/W	R/W	R/W
default	1'b0	1'b0	1'b0	1'b0	1'b0	1'b0	1'b0	1'b0

Bits	Name	R/W	Description
7:6	-	RO	Reserved
5:0	ip1.5-ip1.0	R/W	Group5-0 priority level control. The value of ip0.5-ip0.0 and ip1.5-ip1.0 define the priority level for interrupt group5-group0.

ip1.x	ip0.x	priority level
0	0	level0 (lowest)
0	1	level1
1	0	level2
1	1	level3 (highest)

4.2.18 Ircon (C0H)

Register addr	C0H							
Bit number	7	6	5	4	3	2	1	0
Bit field name	–	–	Iex6	Iex5	Iex4	Iex3	Iex2	iadc
R/W	RO	RO	R/W	R/W	R/W	R/W	R/W	R/W
default	1'b0	1'b0	1'b0	1'b0	1'b0	1'b0	1'b0	1'b0

Bits	Name	R/W	Description
5	Iex6	R/W	External interrupt6 edge flag This bit is set by hardware if detect external interrupt6. And can be cleared by software.
4	Iex5	R/W	External interrupt5 edge flag This bit is set by hardware if detect external interrupt5. And can be cleared by

			software.
3	Iex4	R/W	External interrupt4 edge flag This bit is set by hardware if detect external interrupt4. And can be cleared by software.
2	Iex3	R/W	External interrupt3 edge flag If this bit is set by hardware when detect external int3, it will be kept until software clear.
1	Iex2	R/W	External interrupt2 edge flag If this bit is set by hardware when detect external int2, it will be kept until software clear.
0	Iadc	R/W	A/D converter interrupt request flag This bit is set by hardware if detect A/D converter interrupt. And can be cleared by software.

4.2.19 T2con (C8H)

Register addr	C8H							
Bit number	7	6	5	4	3	2	1	0
Bit field name	-	i3fr	i2fr	-	-	-	-	-
R/W	RO-	R/W	R/W	RO-	RO-	RO-	RO-	RO-
default	1'b0	1'b0	1'b0	1'b0	1'b0	1'b0	1'b0	1'b0

Bits	Name	R/W	Description
7	–	RO	Reserved
6	i3fr	R/W	Select active edge : a. External interrupt 3 b. Compare signal c. Capture signal 0: falling edge 1: rising edge
5	i2fr	R/W	Select active edge : d. External interrupt 2 e. Compare signal f. Capture signal 0: falling edge 1: rising edge
4	–	RO	Reserved
3	–	RO	Reserved
2	–	RO	Reserved

1	–	RO	Reserved
0	–	RO	Reserved

4.2.20 Psw (D0H)

Register addr	D0H							
Bit number	7	6	5	4	3	2	1	0
Bit field name	Cy	Ac	F0	Rs1	Rs0	Ov	-	P
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
default	1'b0	1'b0	1'b0	1'b0	1'b0	1'b0	1'b0	1'b0

Bits	Name	R/W	Description
7	Cy	R/W	Carry flag.
6	Ac	R/W	Auxiliary carry flag for BCD operations (for bit3-0 operation to bit4)
5	F0	R/W	General purpose Flag 0 available for user
4	Rs1	R/W	Register bank select control bit 1, used to select working register bank
3	Rs0	R/W	Register bank select control bit 0, used to select working register bank rs1/rs0 | Bank selected | Location 00 | Bank0 | (00H-07H) 01 | Bank1 | (08H-0FH) 10 | Bank2 | (10H-17H) 11 | Bank3 | (18H-1FH)
2	Ov	R/W	Overflow flag. Signed arithmetic, result overflow(-128-+127) Bit6 operation to bit7 Extension OR bit7 operation to Carry flag
1	-	R/W	reserved
0	p	R/W	Parity flag, affected by hardware to indicate odd/even number of "one" bits in the accumulator, i.e. even parity.

4.2.21 Wdcon (D8H)

Register addr	D8H	
Bit number	7	6:0
Bit field name	wdcon	-
R/W	R/W	RO
default	1'b0	7'b0

Bits	Name	R/W	Description
7	wdcon	R/W	Baud rate generate control 1: baud rate = 2^{smod}xFclk/32x12x(256-th1)

			0: baud rate = 2^{smod}xFclk/64x(2^{10}-s0rel)
6:0	-	RO	Reserved

4.2.22 Acc (E0H)

Acc is the accumulator register. Most instructions use the accumulator to hold the operand.

4.2.23 Md0/md1/md2/md3/md4/md5 (E9H/EAH/EBH/ECH/EDH/EEH)

The operands and results registers for multiply and divide operation.

4.2.24 Arcon (EFH)

Register addr	EFH			
Bit number	7	6	5	4:0
Bit field name	mdef	mdov	slr	sc
R/W	RO	RO	R/W	R/W
default	1'b0	1'b0	1'b0	5'b0

Bits	Name	R/W	Description
7	mdef	R/RC	Error flag Indicates an improperly performed operation(when one of the arithmetic operations has been restarted or interrupted by a new operation)
6	mdov	R/WC	Overflow flag
5	slr	R/W	Shift direction bit 0: shift left 1: shift right
4:0	sc	R/W	Shift counter 5'b00000: normalizing is selected. After normalize, sc[4:0] contain the number of normalizing shifts performed. Others: shift operation is started. The number of shifts performed is determined by the count written to sc[4:0].

4.2.25 B (F0H)

The B register is used during multiply and divide instructions. It can also be used as a scratch-pad register to hold temporary data.

4.2.26 Probel_sel (FBH)

R80515 and JTAG modules probe signals select

Bits	Name	R/W	Description
7:5	-	RO	Reserved
4:0	probel_sel	R/W	R80515 and JTAG modules probe signals select, select low 8bit probe signal 5'b00000: pc[7:0] 5'b00001: pc[15:8] 5'b00010: romdatai[7:0] 5'b00011: xramaddr[7:0]

			5'b00100: xramaddr[15:8] 5'b00101: xramdatai[7:0] 5'b00110: xramdatao[7:0] 5'b00111: iramaddr[7:0] 5'b01000: iramdatai[7:0] 5'b01001: iramdatao[7:0] 5'b01010: sfraddr[7:0] 5'b01011: sfrdatai[7:0] 5'b01100: sfrdatao[7:0] 5'b01101: {2'b0, intreq, intack, intret, xramen, iramen, sfren} 5'b01110: instr[7:0] 5'b01111: acc[7:0] 5'b10000: {a5_break, clkmcu_en, sc_sel_reg[3:0], ice_en, romen_ice} 5'b10001: {tck, tms, tdi, tdo, capture_dr_tap, shift_dr_tap, step_req, run_req}

4.2.27 Probeh_sel (FCH)

R80515 and JTAG modules probe signals select

Bits	Name	R/W	Description
7:5	-	RO	Reserved
4:0	probeh_sel	R/W	R80515 and JTAG modules probe signals select, select high 8bit probe signal 5'b00000: pc[7:0] 5'b00001: pc[15:8] 5'b00010: romdatai[7:0] 5'b00011: xramaddr[7:0] 5'b00100: xramaddr[15:8] 5'b00101: xramdatai[7:0] 5'b00110: xramdatao[7:0] 5'b00111: iramaddr[7:0] 5'b01000: iramdatai[7:0] 5'b01001: iramdatao[7:0] 5'b01010: sfraddr[7:0] 5'b01011: sfrdatai[7:0] 5'b01100: sfrdatao[7:0] 5'b01101: {2'b0, intreq, intack, intret, xramen, iramen, sfren} 5'b01110: instr[7:0] 5'b01111: acc[7:0] 5'b10000: {a5_break, clkmcu_en, sc_sel_reg[3:0], ice_en, romen_ice} 5'b10001: {tck, tms, tdi, tdo, capture_dr_tap, shift_dr_tap, step_req, run_req}

5 Function

5.1 Instruction list

5.1.1 1 cycle

Instruction	Description	Code	Byte	Cycles
NOP	No operation	00	1	1
RR A	Rotate accumulator right	03	1	1
INC A	Increment accumulator	04	1	1
RRC A	Rotate accumulator right through carry	13	1	1
DEC A	Decrement accumulator	14	1	1
RL A	Rotate accumulator left	23	1	1
ADD A，@Ri	Add indirect RAM to accumulator	26-27	1	1
ADD A，Rn	Add register to accumulator	28-2F	1	1
RLC A	Rotate accumulator left through carry	33	1	1
ADDC A，@Ri	Add indirect RAM to accumulator with carry flag	36-37	1	1
ADDC A，Rn	Add register to accumulator with carry flag	38-3F	1	1
ORL A，@Ri	OR indirect RAM to A	46-47	1	1
ORL A，Rn	OR register to accumulator	48-4F	1	1
ANL A，@Ri	AND indirect RAM to A	56-57	1	1
ANL A，Rn	AND register to accumulator	58-5F	1	1
XRL A，@Ri	Exclusive OR register to accumulator	66-67	1	1
XRL A，Rn	Exclusive OR register to accumulator	68-6F	1	1
SUBB A，@Ri	Subtract indirect RAM from accumulator with borrow	96-97	1	1
SUBB A，Rn	Subtract register from accumulator with borrow	98-9F	1	1
INC DPTR	Increment data pointer	A3	1	1
CPL C	Complement carry flag	B3	1	1
CLR C	Clear carry flag	C3	1	1
SWAP A	Swap nibbles within the accumulator	C4	1	1

SETB C	Set carry flag	D3	1	1
DA A	Decimal adjust accumulator	D4	1	1
CLR A	Clear accumulator	E4	1	1
MOV A，@Ri	Move indirect RAM to accumulator	E6-E7	1	1
MOV A，Rn	Move register to accumulator	E8-EF	1	1
CPL A	Complement accumulator	F4	1	1

5.1.2 2 cycles

Instructions	Description	Code	Byte	Cycles
INC @Ri	Increment indirect RAM	06-07	1	2
INC Rn	Increment register	08-0F	1	2
DEC @Ri	Decrement indirect RAM	16-17	1	2
DEC Rn	Decrement register	18-1F	1	2
ADD A，#data	Add immediate data to accumulator	24	2	2
ADD A，direct	Add direct byte to accumulator	25	2	2
ADDC A，#data	Add immediate data to A with carry flag	34	2	2
ADDC A，direct	Add direct byte to A with carry flag	35	2	2
JC rel	Jump if carry flag is set	40	2	2
ORL A，#data	OR immediate data to A	44	2	2
ORL A，direct	OR direct byte to A	45	2	2
JNC rel	Jump if carry flag is not set	50	2	2
ANL A，#data	AND immediate data to A	54	2	2
ANL A，direct	AND direct byte to A	55	2	2
JZ rel	Jump if accumulator is Zero	60	2	2
XRL A，#data	Exclusive OR immediate data to A	64	2	2

XRL A，direct	Exclusive OR direct byte to A	65	2	2
JNZ rel	Jump if accumulator is not Zero	70	2	2
ORL C，bit	OR direct bit to carry flag	72	2	2
JMP @A+DPTR	Jump indirect relative to the DPTR	73	1	2
MOV A，#data	Move immediate data to accumulator	74	2	2
MOV @Ri，#data	Move immediate data to indirect RAM	76-77	2	2
MOV Rn，#data	Move immediate data to register	78-7F	2	2
SJMP rel	Short jump	80	2	2
ANL C，bit	AND direct bit to carry flag	82	2	2
MOV direct，@Ri	Move indirect RAM to direct byte	86-87	2	2
MOV direct，Rn	Move register to direct byte	88-8F	2	2
SUBB A，#data	Subtract immediate data from A with borrow	94	2	2
SUBB A，direct	Subtract direct byte from A with borrow	95	2	2
ORL C，/bit	OR complement of direct bit to carry	A0	2	2
MOV C，bit	Move direct bit to carry flag	A2	2	2
ANL C，/bit	AND complement of direct bit to carry	B0	2	2
XCH A，@Ri	Exchange indirect RAM with accumulator	C6-C7	1	2
XCH A，Rn	Exchange register with accumulator	C8-CF	1	2
POP direct	Pop direct byte from stack	D0	2	2
XCHD A，@Ri	Exchange low-order nibble of indirect RAM with accumulator	D6-D7	1	2
DJNZ Rn，rel	Decrement register and jump if not zero	D8-DF	2	2
MOVX A，@DPTR	Move external RAM(16bit address) to accumulator	E0	1	2

MOVX A，@Ri	Move external RAM(8bit address) to accumulator	E2-E3	1	2
MOV A，direct	Move direct byte to accumulator	E5	2	2
AJMP addr11	Absolute jump	Xxx01	2	2
MOVX @DPTR，A	Move A to external RAM(16bit address)	F0	1	2
MOVX @Ri，A	Move A to external RAM(8bit address)	F2-F3	1	2
MOV direct，A	Move accumulator to direct byte	F5	2	2
MOV @Ri，A	Move accumulator to indirect RAM	F6-F7	1	2
MOV Rn，A	Move accumulator to register	F8-FF	1	2

5.1.3 3 cycles

Instruction	description	Code	Byte	Cycles
LJMP addr16	Long jump	02	3	3
INC direct	Increment direct byte	05	2	3
JBC bit，rel	Jump if direct bit is set and clear bit	10	3	3
LCALL addr16	Long subroutine call	12	3	3
DEC direct	Decrement direct byte	15	2	3
JB bit，rel	Jump if direct bit is set	20	3	3
RET	From subroutine	22	1	3
JNB bit，rel	Jump if direct bit is not set	30	3	3
RETI	From interrupt	32	1	3
ORL direct，A	OR accumulator to direct byte	42	2	3
ANL direct，A	AND accumulator to direct byte	52	2	3
XRL direct，A	Exclusive accumulator to direct byte	62	2	3
MOV direct，#data	Move immediate data to direct byte	75	3	3

MOVC A，@A+PC	Move code byte relative to PC to accumulator	83	1	3
MOV direct1，direct2	Move one direct byte to another direct byte	85	3	3
MOV DPTR，#data16	Load data pointer with 16-bit constant	90	3	3
MOV bit，C	Move carry flag to direct bit	92	2	3
MOVC A，@A+DPTR	Move code byte relative to DPTR to accumulator	93	1	3
MOV @Ri，direct	Move direct byte to indirect RAM	A6-A7	2	3
MOV Rn，direct	Move direct byte to register	A8-AF	2	3
CPL bit	Complement direct bit	B2	2	3
CJNE A，#data rel	Compare immediate data with A and jump if not equal	B4	3	3
CJNE A，direct rel	Compare direct byte with A and jump if not equal	B5	3	3
CJNE @Ri，#data rel	Compare indirect RAM with immediate data and jump if not equal	B6-B7	3	3
CJNE Rn，#data rel	Compare register with immediate data and jump if not equal	B8-BF	3	3
PUSH direct	Push direct byte onto stack	C0	2	3
CLR bit	Clear direct bit	C2	2	3
XCH A，direct	Exchange direct byte with accumulator	C5	2	3
SETB bit	Set direct bit	D2	2	3
DJNZ direct，rel	Decrement direct byte and jump if not zero	D5	3	3
ACALL addr11	Absolute subroutine call	Xxx11	2	3

5.1.4 4 cycles

Instruction	description	Code	Byte	Cycles
ORL direct, #data	OR immediate data to direct byte	43	3	4
ANL direct，#data	AND immediate data to direct byte	53	3	4
XRL direct，#data	Exclusive OR immediate data to direct byte	63	3	4

5.1.5 5 cycles

Instruction	description	Code	Byte	Cycles
DIV AB	Divide A by B	84	1	5
MUL AB	Multiply A and B	A4	1	5

5.2 Interrupt Service

The R80515 provides 13 interrupt sources with four priority levels. Each source has its own request flag located in SFR tcon/ircon/s0con/s1con. Each interrupt has individually enable or disable bit in SFR ien0, ien1 and ien2.
When the interrupt occurs, the engine will vector to the predetermined address. Once interrupt service has begun, it can be interrupted only by a higher priority interrupt. The interrupt service is terminated by an RETI instruction. When an RETI is performed, the processer will return to the instruction that would have been next when interrupt occurred.
When the interrupt condition occurs, the processor will also indicate this by setting a flag bit. This bit is set regardless of whether the interrupt is enabled or disabled. Each interrupt flag is sampled once per cycle and polled by hardware. If the sample indicates a pending interrupt when the interrupt is enabled, then interrupt request flag is set. On the next instruction cycle, the interrupt will be acknowledged by hardware forcing an LCALL to appropriate vector address.
Interrupt response will require a varying amount of time depending on the state of microcontroller when the interrupt occurs. If microcontroller is performing an interrupt service with equal or greater priority, the new interrupt will be invoked. In other cases, the response time depends on current instruction. The fastest possible response to an interrupt is 4 cycles which includes one machine cycle for detecting the interrupt and 3 cycles for perform the LCALL.

5.2.1 Interrupt sources and vectors

Interrupt sources	Description	Interrupt vector
ie0-External interrupt 0	Enable: ien0.7 & ien0.0(SW ctrl) Type control: tcon.0(SW ctrl) Flag: tcon.1(HW ctrl)	0003H
tf0-Timer0 interrupt	Enable: ien0.7 & ien0.1(SW ctrl) Flag: tcon.5(HW ctrl)	000BH
ie1-External interrupt1	Enable: ien0.7 & ien0.2(SW ctrl) Type control: tcon.2(SW ctrl) Flag: tcon.3(HW ctrl)	0013H
tf1-Timer1 interrupt	Enable: ien0.7 & ien0.3(SW ctrl) Flag: tcon.7(HW ctrl)	001BH
ri0/ti0- Serial 0 transmit/receive interrupt	Enable: ien0.7 & ien0.4(SW ctrl) Flag: s0con.0/s0con.1(HW ctrl)	0023H
ri1/ti1- Serial 1transmit/receive interrupt	Enable: ien0.7 & ien2.0(SW ctrl) Flag: s1con.0/s1con.1(HW ctrl)	0083H
iadc-A/D converter interrupt	Enable: ien0.7 & ien1.0(SW ctrl) Flag: ircon.0(HW ctrl)	0043H
iex2-External interrupt 2	Enable: ien0.7 & ien1.1(SW ctrl) Type control: t2con.5 (SW ctrl) Flag: ircon.1(HW ctrl)	004BH

iex3-External interrupt 3	Enable: ien0.7 & ien1.2(SW ctrl) Type control: t2con.6(SW ctrl) Flag: ircon.2(HW ctrl)	0053H
iex4-External interrupt 4	Enable: ien0.7 & ien1.3(SW ctrl) Flag: ircon.3(HW ctrl)	005BH
iex5-External interrupt 5	Enable: ien0.7 & ien1.4(SW ctrl) Flag: ircon.4(HW ctrl)	0063H
iex6-External interrupt 6	Enable: ien0.7 & ien1.5(SW ctrl) Flag: ircon.5(HW ctrl)	006BH

5.2.2 Priority level structure

The priority level groups as follow:

External interrupt 0	Serial channel 1 interrupt	A/D converter interrupt	group0
Timer 0 interrupt	-	External interrupt 2	group1
External interrupt 1	-	External interrupt 3	group2
Timer 1 interrupt	-	External interrupt 4	group3
Serial 0 interrupt	-	External interrupt 5	group4
	-	External interrupt 6	group5

Each group of interrupt sources can be programmed individually to one of four priority levels by setting or clearing one bit in the SFR ip0 and ip1. If requests of the same priority level will be received simultaneously, an internal polling sequence determines which request is serviced first.

The polling sequence as follows (from up to down):

External interrupt0	
Serial 1 interrupt	
A/D converter interrupt	
Timer 0 interrupt	
External interrupt 2	
External interrupt 1	
External interrupt 3	
Timer 1 interrupt	
External interrupt 4	
Serial 0 interrupt	
External interrupt 5	
Timer 2 interrupt	
External interrupt	polling sequence

5.2.3 Interrupt service for SD control

SD control use external interrupt2 and external interrupt3. And all this two interrupt use falling edge, and external interrupt2 is used for command interrupt, external interrupt3 is used for data interrupt. If firmware doesn't set priority control register (ip0 and ip1), the polling sequence is external interrupt2 first and external interrupt3 second. And MCU can support one same or some different interrupt sources pending when this interrupt is processing.

The MCU of SD control supports field maintenance function. And this function will be enabled every interrupt. When interrupt detect, MCU will register current values of PSW/ACC/B/DPTR/DPTR1, the contents of PSW/ACC/B/DPTR/DPTR1 will be reloaded by registered values when interrupt return. This function is used to avoid modifying during interrupt process.

There are another function for SD project. This function is translating LJMP command to JMP command. This function is controlled by Ljmpc.0 SFR. If receive LJMP addr16 command, and ljmpc.0 is set, the program memory address and PC will jump to cmd_index*3+addr16. Cmd_index is the "command index" field of SD Command Format.

5.3 Serial interface 0 and 1

The serial0 and 1 have two separate registers, one Transmit Buffer and one Receive Buffer.

Writing data to the Special Function Register S0BUF or S1BUF sets this data in serial output buffer and starts the transmission. Reading from the S0BUF or S1BUF reads data from the serial receive buffer after detecting the receive-completed-interrupt. The serial port can simultaneously transmit and receive data. It can also buffer 1 byte at receive, which prevents the receive data from being lost if the CPU reads the first byte before transmission of the second byte is completed.

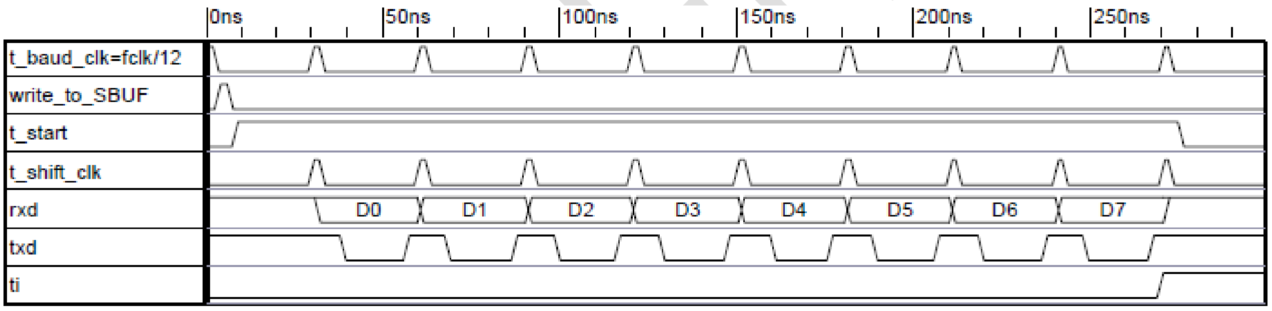
5.3.1 Serial interface 0 mode

Serial interface 0 has 4 modes by setting s0con[7:6], refer to the description of s0con.

5.3.1.1 Mode0

5.3.1.1.1 Transmit

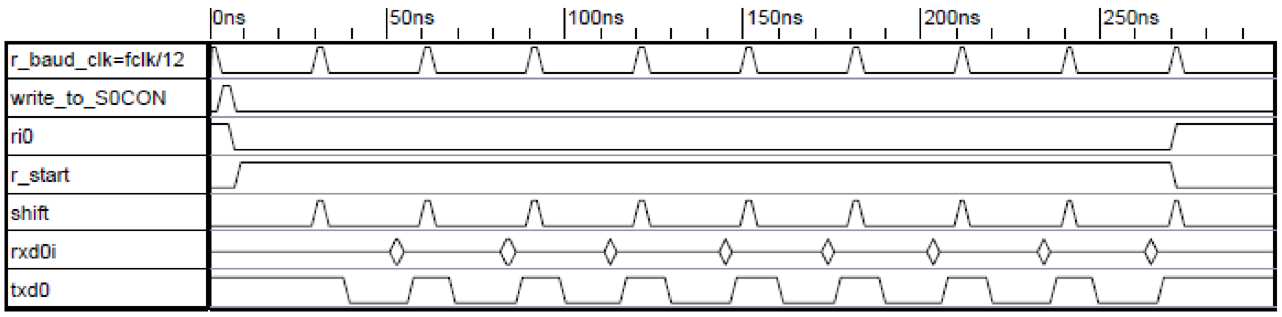
Set s0con[7:6]=2'b00, if write values to s0buf, the transmit will be started, and the value in s0buf are transmitted with LSB first at rxd0o. Baud rate is fixed at 1/12 of crystal frequency. Txd0 outputs the shift clock. S0con.1 will be set after transmit completed.



Transmit mode0 for serial0

5.3.1.1.2 Receive

Starts receiving if s0con.0 from 1 to 0 and s0con.4=1. The received data from rxd0i will be written into s0buf. Txd0 outputs the shift clock.

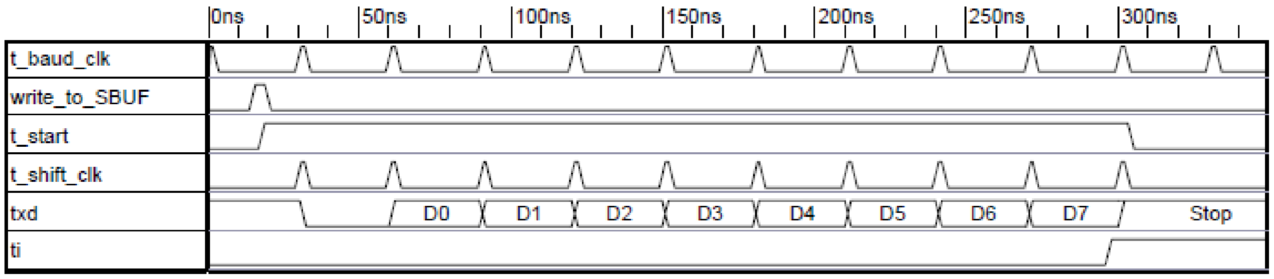


Receive mode0 for serial0

5.3.1.2 Mode1

5.3.1.2.1 Transmit

Set s0con[7:6]=2'b01, if write value s to s0buf, the transmit will be started, and the value in s0buf are transmitted with LSB first at txd0. 10 bits are transmitted: a start bit (always 0), 8 data bits (LSB first), and a stop bit (always 1).



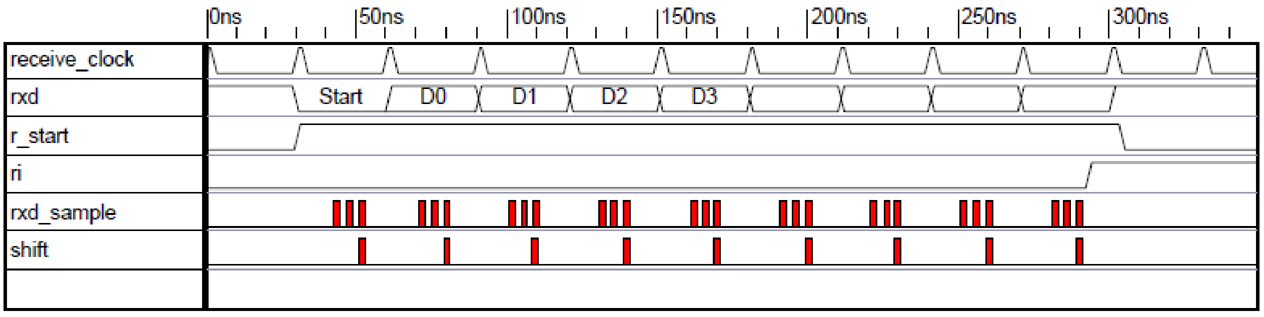
Transmit mode1 for serial0

5.3.1.2.2 Receive

Rxd0i as input

On receive, a start bit synchronizes the transmission, 8 data bits are available by reading s0buf, s0con.4=0 and s0con.2=1 means stop.

Internal baud rate generator or timer1 can be bused to specify baud rate.



Receive mode1 for serial0

Baud rate generation: (based on wdcon.7)

Wdcon.7=0: baud rate = 2^{smod}xFclk/32x12x(256-th1)

Wdcon.7=1: baud rate = 2^{smod}xFclk/64x(2^{10}-s0rel)

S0rel is 10bit width, s0rel=s0relh.[1:0] + s0rell.[7:0].

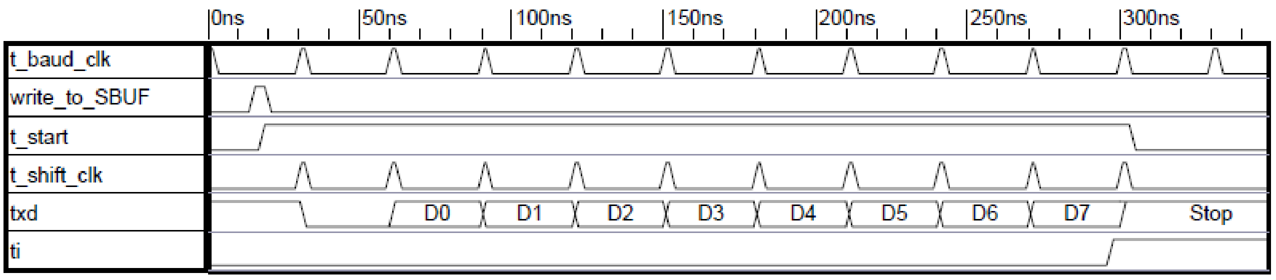
5.3.1.3 Mode2

5.3.1.3.1 Transmit

Set s0con[7:6]=2'b10, if write value to s0buf, the transmit will be started, and the value in s0buf are transmitted with LSB first at txd0.

No external shift clock is used

11 bits are transmitted: a start bit (0), 8 bits data (LSB first), a programmable 9th bit (s0con.3), a stop bit(1).

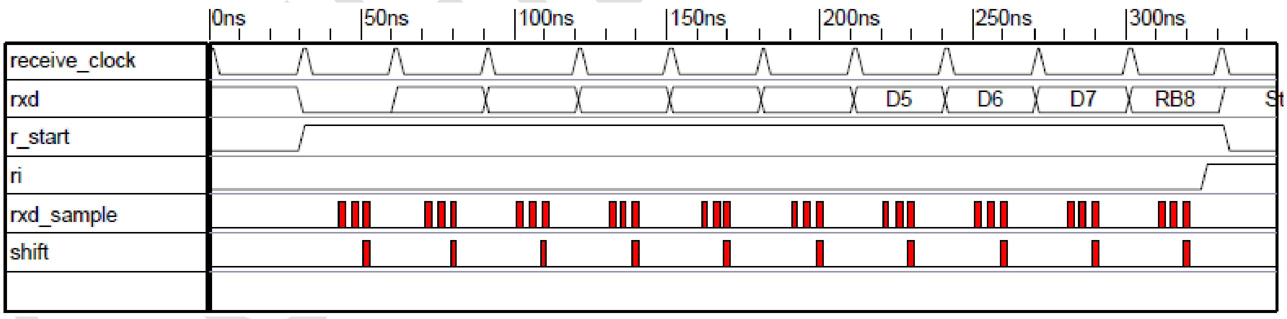


Transmit mode2 for serial0

5.3.1.3.2 Receive

Rxd0i as input

On receive: a start bit synchronizes the transmission, 8 bits data, 9th bit is written to s0con.2, a stop bit.



Receive mode2 for serial0

Baud rate is fixed at 1/32 or 1/64 of oscillator frequency based on pcon.7.

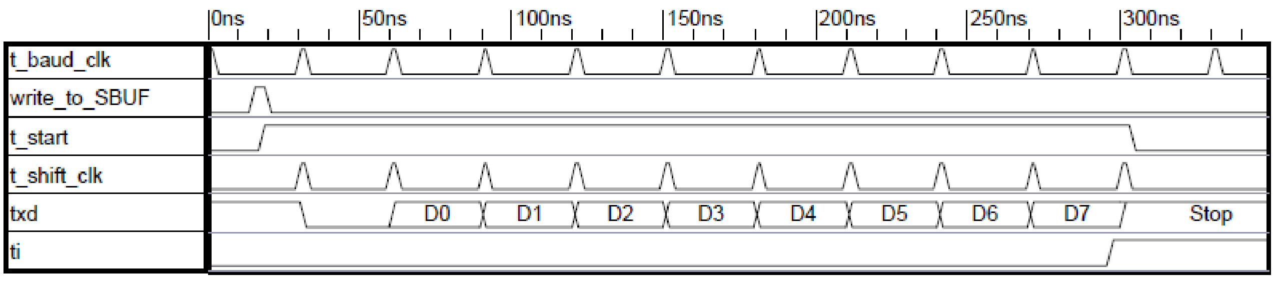
5.3.1.4 Mode3

5.3.1.4.1 Transmit

Txd0 as output

No external shift clock is used

11 bits are transmitted: a start bit (0), 8 bits data (LSB first), a programmable 9th bit (s0con.3), a stop bit (1).



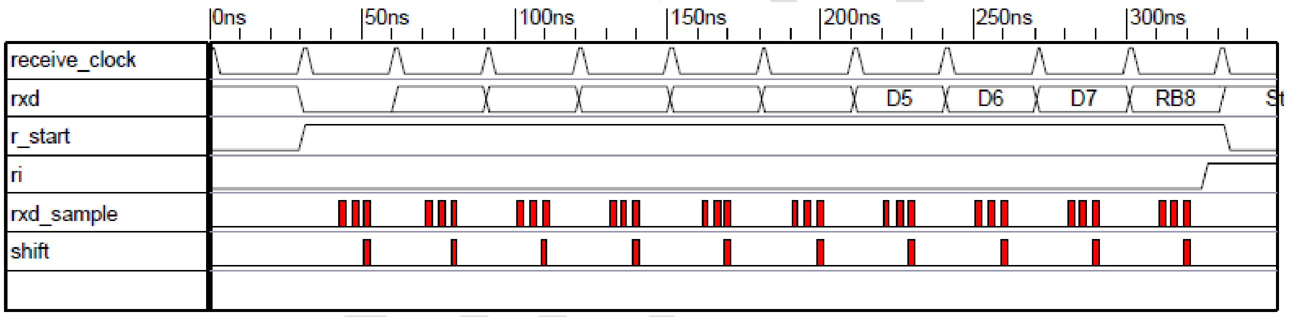
Transmit mode3 for serial0

5.3.1.4.2 Receive

Rxd0i as input

On receive: a start bit synchronizes the transmission (0), 8 bits data, 9th bit is written to s0con.2, a stop bit.

Internal baud rate generator or timer1 can be used to specify baud rates



Receive mode3 for serial0

Baud rate generation: (based on wdcon.7)

Wdcon.7=0: baud rate = 2^{smod}xFclk/32x12x(256-th1)

Wdcon.7=1: baud rate = 2^{smod}xFclk/64x(2^{10}-s0rel)

S0rel is 10bit width, s0rel=s0relh.[1:0] + s0rell.[7:0].

5.3.2 Serial interface 1 modes

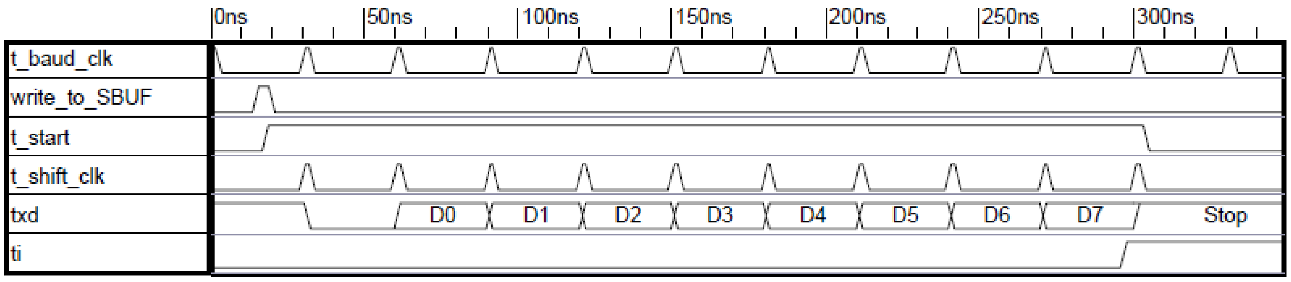
5.3.2.1 Mode A

5.3.2.1.1 Transmit

Txd1 as output

No external shift clock is used

11 bits are transmitted: a start bit (0), 8 bits data (LSB first), a programmable 9th bit (s1con.3), a stop bit (1).

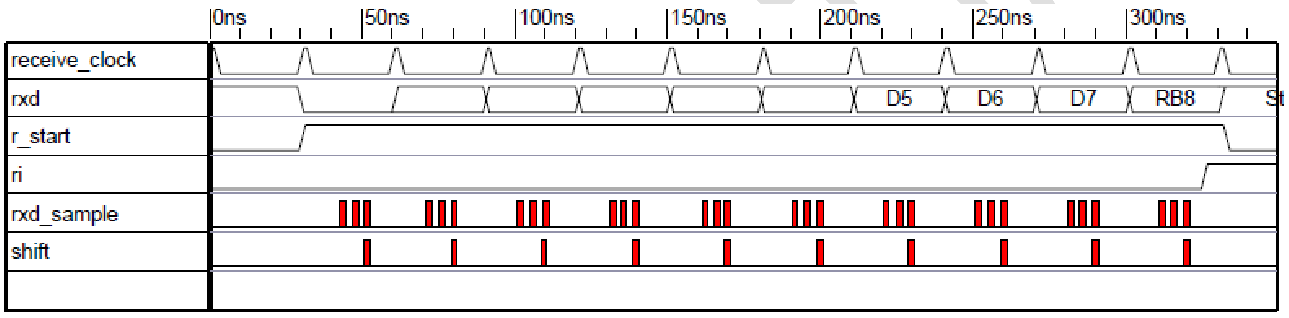


Transmit modeA for serial1

5.3.2.1.2 Receive

Rxd1i as input

On receive: a start bit synchronizes the transmission, 8 bits data, 9th bit is written to s0con.2, a stop bit.



Receive modeA for serial1

Baud rate:

Fclk/32x(2^{10}-s0rel)

S0rel is 10bit width, s0rel=s0relh.[1:0] + s0rell.[7:0].

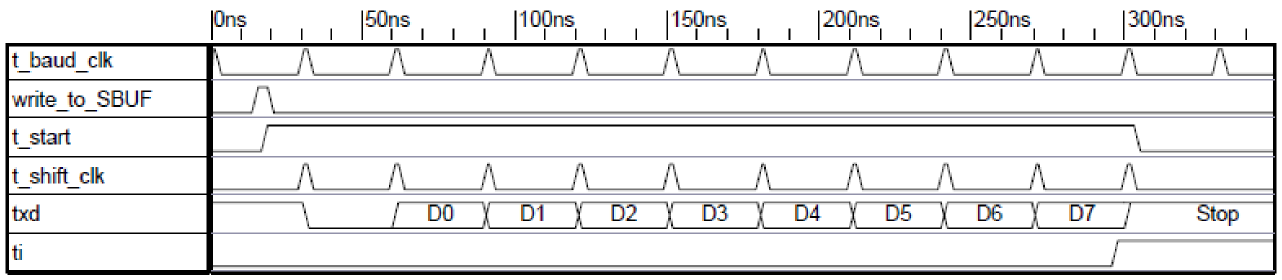
5.3.2.2 Mode B

5.3.2.2.1 Transmit

Txd1 as output

No external shift clock is used

10 bits are transmitted: a start bit (0), 8 bits data (LSB first), a stop bit (1).



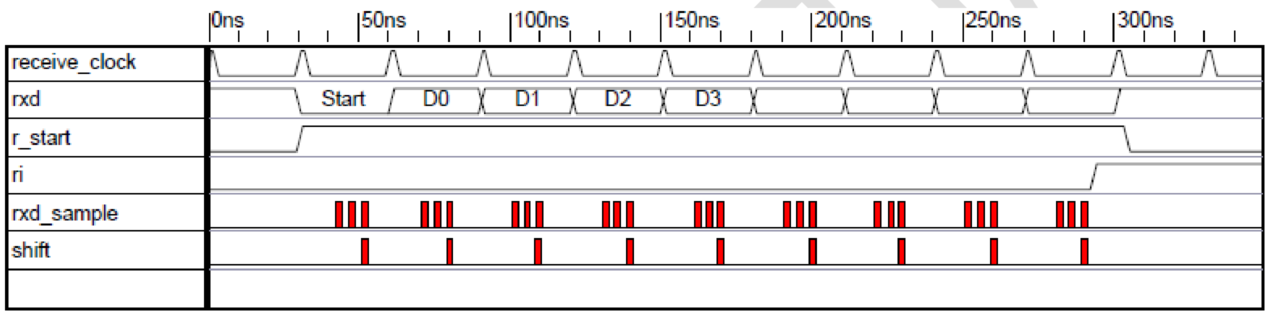
Transmit modeB for serial1

5.3.2.2.2 Receive

Rxd1i as input

On receive: a start bit synchronizes the transmission, 8 bits data, a stop bit (s1con.5=0, s1con.2=1)..

Internal baud rate generator can be used to specify baud rates.



Receive modeB for serial1

Baud rate:

Fclk/32x(2^{10}-s0rel)

S0rel is 10bit width, s0rel=s0relh.[1:0] + s0rell.[7:0].

5.3.3 Sample data during receive

5.3.4 Multiprocessor Communication of Serial Interface 0 and 1

The feature of receiving 9 bits in mode 2 and mode 3 of serial 0 or in mode A of serial 1 can be used for multiprocessor communication. In this case, the slave processors have bit sm20 in s0con or sm21 in s1con set to 1. When the master processor outputs slave's address, it sets the 9th bit (s0con.3/s1con.3) to 1, causing a serial port receive interrupt in all the slaves. The slave processors compare the received byte with their network address. If there is a match, the addressed slave will clear sm20 or sm21 and receive the rest of the message, while other slaves will leave sm20 or sm21 bit unaffected and ignore this message. After addressing the slave, the host will output the rest of the message with the 9th bit set to 0, so no serial port receive interrupt will be generated in unselected slaves.

5.4 Multiplication Division unit

5.4.1 Operation phases of the MDU

There are three phases of the operation of the MDU:

5.4.1.1 First phase: loading the mdx registers

The loading mdx FSM:

Current state	Conditions	Next state	Signals	Operation

ST0	Write md0	ST1		MDU nop
	Write arcon	ST7		
ST1	Write md0	ST13		MDU reset
	Write md1	ST2		
	Write md4	ST10		
	Write arcon	ST7		
ST2	Write md0	ST13		MDU nop
	Write md2	ST3		
	Write md4	ST8		
	Write arcon	ST7		
ST3	Write md0	ST13		MDU nop
	Write md3	ST4		
	Write arcon	ST7		
ST4	Write md0	ST13		MDU nop
	Write md4	ST5		
	Write arcon	ST7		
ST5	Write md0	ST13		MDU nop
	Write md5	ST6		
	Write arcon	ST7		
ST6	Write md0	ST13		MDU div32
	Write md1-md5	ST14		
	DIV32 operation END & read md5	ST0		
ST7	Write md0	ST13		MDU shift
	Write md1-md5	ST14		
	Shift operation END & read md3	ST0		
ST8	Write md0	ST13		MDU nop
	Write md5	ST9		
	Write arcon	ST7		
ST9	Write md0	ST13		MDU div16
	Write md1-md5	ST14		
	Div16 operation END & read md5	ST0		
ST10	Write md0	ST13		MDU nop
	Write md1	ST11		
	Write arcon	ST7		
ST11	Write md0	ST13		MDU nop
	Write md5	ST12		
	Write arcon	ST7		

ST12	Write md0	ST13		MDU multiplication
	Write md1-md5	ST14		
	Mul operation END & read md3	ST0		
ST13		ST1	Setmdef=1 (arcon.7=1)	MDU reset
ST14		ST0	Setmdef=1 (arcon.7=1)	MDU reset

For any operation, if not write in sequence, the MDEF flag will be set (arcon.7).
Conclusion:
The default and power on state is ST0,if write SFR (md0 or arcon), the state will transfer to another.
ST6: div32;;;;;;ST7: shift;;;;;;ST9: div16;;;;;;ST12: multiplication;;;;
Correct div32 flow:
Write md0-md5 in sequence
The state will transfer to ST6, and wait div32 operation end.
If div32 finish and read md5, the state will return to ST0
Correct shift flow:
Write arcon to start shift,
The state will transfer to ST7, and wait shift operation end
If shift finish and read md3, the state will return to ST0
Correct div16 flow:
Write md0,md1,md4,md5 in sequence
The state will transfer to ST9, and wait div16 end
If div16 finish and read md5, the state will return to ST0
Correct multiplication flow:
Write md0, md4, md1, md5 in sequence
The state will transfer to ST12, and wait multiplication end
If multiplication finish and read md3, the state will return to ST0
After start the other operation but not shift, the shift can be started by write arcon
If write md0 again after write md0, the state will transfer to ST13, The mdef will be set at ST13, and the state will transfer to ST1, and wait the next operation: write md1/md4/arcon
During wait some operation end state, ST6 for div32, ST7 for shift, ST9 for div16, ST12 for multiplication, if write md1-md5, the state will transfer to ST14 which will transfer to ST1 and generate mdef.

5.4.1.2 Second phase: executing calculation

The arithmetic operation FSM:

Current state	Conditions	Next state	Signals	Operation
ST0	Mdu reset	ST12	Count_sel=2’b01 Opend=1	Md30_sel=nop
		ST0		
ST1	Arcon[4:0]=5’b00000 & md3.7=1	ST8	Count_sel=2’b01	Md30_sel=nop
	Arcon[4:0]=5’b00000 &	ST7		

	md3.7=0			
	arcon[4:0]!=5’b00000 & arcon[5]	ST5		
	arcon[4:0]!=5’b00000 & !arcon[5]	ST6		
ST2	Counter_st=5’b10010	ST9	Counter_sel=2’b11	Md30_sel=MUL
	Counter_st!=5’b10010	ST2		
ST3	Counter_st=5’b10100	ST10	Counter_sel=2’b11	Md30_sel=DIV16
	Counter_st!=5’b10100	ST3		
ST4	Counter_st=5’b00100	ST10	Counter_sel=2’b11	Md30_sel=DIV32
	Counter_st!=5’b00100	ST4		
ST5	Counter_st=5’b00001 | Counter_st=5’b00010 | Counter_st=5’b00000	ST0	Counter_sel=2’b11	Md30_sel=Shift
	Others	ST5		
ST6	Counter_st=5’b00001 | Counter_st=5’b00010 | Counter_st=5’b00000	ST0	Counter_sel=2’b11	Md30_sel=Shift
	Others	ST6		
ST7	Counter_st=5’b00010 | md3.6 | md3.5	ST11	Counter_sel=2’b11	Md30_sel=Normalizing
	Others	ST7		
ST8		ST0	Counter_sel=2’b10 Setmdov=1	Md30_sel=NOP
ST9		ST0	Counter_sel=2’b10 Setmdov=(|md3) | (|md2)	Md30_sel=NOP
ST10		ST0	Counter_sel=2’b10 Setmdov=(|Md4=0 & |md5=0) ? 1’b1 : 1’b0	Md30_sel=NOP
ST11		ST0	Counter_sel=2’b10 Ld_sc=1//load sc[4:0]	Md30_sel=NOP
ST12(the state after mdu operation reset)	Mdu operation=shift	ST1	Counter_sel=2’b10	Md30_sel=NOP
	Mdu operation=mul	ST13		
	Mdu operation=div16	ST3		
	Mdu operation=div32	ST4		
ST13		ST2	Counter_sel=2’b10	Md30_sel=md3 reset

Conclusion:

There are 6 different operations: MUL(ST2),DIV16(ST3),DIV32(ST4),SHIFT LEFT(ST6),SHIFT RIGHT(ST5),NORMALIZE(ST7)

Normalization is started by writing arcon[4:0]=5’b0 & md3[7]=0

Shift calculation flow:

After write md0, the calculation FSM will transfer to ST12 and wait the load operation end. If the load operation finished correctly, the state will transfer to different next state base on the operation type.

The state will transfer to ST1 for shift operation

If shift right (arcon[5]=1), then next state is ST5. If shift left (arcon[5]=0), then next state is ST6

5.4.1.3 Third phase: reading the result from mdx

The last read (from md5 for division and md3 for multiplication, shift and normalizing) determines the end of the whole calculation.

5.4.2 MDEF flag

The MD error flag mechanism is automatically enabled (not means set) with the first write to md0 and disabled (not means reset) with the final read instruction from md3 (for multiplication or shift/normalizing) or md5 (division) in phase three operation.

The set conditions:

Write access to mdx registers in phase two which means re-write mdx after load mdx but before finish operation.

After error flag enabled, but read (not write) mdx registers

The reset conditions:

Phase two finished and read access md3 or md5

5.4.3 MDOV flag

The set conditions:

Division by zero

Multiplication with a result greater than 0000FFFFH

Start of normalizing if md3[7]=1

The reset condition:

Write access to md0

5.4.4 Normalizing

If write 5'b00000 to arcon[4:0], the normalizing operation is started. All reading zeros of integer variables in registers md0 to md3 are removed by shift left operations. The whole operation is completed when the MSB of md3 register contains a '1'. After normalizing, arcon[4:0] contains the number of shift left operations, which were done.

5.4.5 Shifting

Arcon.5 contains the shift direction, and the arcon[4:0] is the shift count (which must not be 0). During shift, zeros come into left or right end of the registers md0 or md3, respectively.

5.4.6 Conclusion

There are 5 different operations are implemented at MDU module. The correct operation flow as follow:

DIV32 flow:

Write md0-md5 in sequence

Wait DIV32 operation end, the max clock number is 50 for DIV32 operation.

Then you can read the result, the quotient is in md0-md3, and the remainder is in md4 and md5. After read md5, the correct DIV32 is finished

DIV16 flow:

Write md0,md1,md4,md5 in sequence

Wait DIV16 operation end, the max clock number is 34 for DIV16 operation.

Then you can read the result, the quotient is in md0-md1, and the remainder is in md4 and md5. After read md5, the correct DIV16 is finished

Shift flow:

Write the shifted data into md0-md3

Write arcon to start shift, arcon[5]=1 indicates shift right; arcon[5]=1 indicates shift left. And the shift number load in arcon[4:0].

Wait shift operation end, the max clock number is 33 for Shift operation.

Then you can read the result from md0-md3. After read md3, the correct shift is finished

Normalization flow:

Write the normalized data into md0-md3

Write arcon[4:0]=5'b0 to start normalization

Wait normalization operation end, the max clock number is 34 for Normalization operation.

Then you can read the result from md0-md3. After read md3, the correct normalization is finished. And the number of normalizing shifts performed in arcon[4:0].

Multiplication flow:

Write md0, md4, md1, md5 in sequence

Wait Multiplication end, the max clock number is 18 for MUL operation.

Then you can read the result from md0-md3. After read md3, the correct Multiplication is finished.

Error or special conditions:

After start the other operation but not shift/normalizing, the shift/normalizing can be started by write arcon at any time

You can reset the operation by writing md0 again, but the mdef (arcon[7]) will be set.

During wait one operation end, if write md1-md5, the operation also can be reset, and the mdef will be set.

5.5 Power Management

R80515 serves two power management modes IDLE and STOP. The control SFR is pcon.

5.5.1 Idle mode

Setting the idle bit (pcon.0) invokes the Idle mode. Internal clocks and peripherals keep running in Idle mode, but clkcpuo will stop. Clkcpuo is the clock of access SFR/IRAM/XRAM/XROM. The MCU will exit the Idle state with any interrupts or reset.

5.5.2 Stop mode

Setting the stop bit (pcon.1) invokes the Stop mode. All internal clocks will turn off in Stop mode.

The MCU will exit the Stop state with no-clocked external interrupt or a reset condition, and internally generated interrupts (timer, serial port, watchdog..) are not useful since they require clock activity.

5.6 Timer 0 and 1

The R80515 has two 16-bit timer/counter registers: Timer0 and Timer1. All can be configured for counter or timer operations.

In timer mode, the register is incremented every machine cycle, which means that it counts up after every 12 periods of the pin clk.

In counter mode, the register is incremented when the falling edge is observed at the corresponding input pin t0 or t1. Since it takes 2 cycles to recognize a 1-to-0 event, the maximum input count rate is 1/2 of the oscillator frequency. There are no restrictions on the duty cycles however, to ensure proper recognition of 0 or 1 state, an input should be stable for at least 1 cycle.

The relative SFR is tmod/tcon.

5.7 Watchdog timer

The watchdog timer is a 16-bit counter that is incremented once every 24 or 384 clock cycles. After an external reset the watchdog is disabled and all registers are set to zeros.

The watchdog consists of 16-bit counter watchdog timer, the reload register is wdtrel.

5.7.1 Start procedure

There are two ways to start watchdog timer.

Hardware automatic start.

This method is based on examining the level of pin swd during active internal rst signal. When this condition is met, the watchdog will start running automatically with default setting (all registers set to zeros).

Programmer start

If hardware can't detect the right level of swd during rst signal, a programmer can start the watchdog later. It will occur when signal swdt (ien1.6) becomes active.

Once the watchdog timer is started it can't be stopped unless rst signal becomes active.

When watchdog timer enters the value of 16'h7FFC, asynchronous wdts signal will be active. The signal wdts (internal signal) sets the ip0.6 and requests reset state. The wdts is cleared either by rst signal or by changing of the state of the value of watchdog timer.

5.7.2 Refresh the watchdog timer

The watchdog timer must be refreshed regularly to prevent reset request signal from becoming active.

The refresh step:

Set wdt (ien0.6)

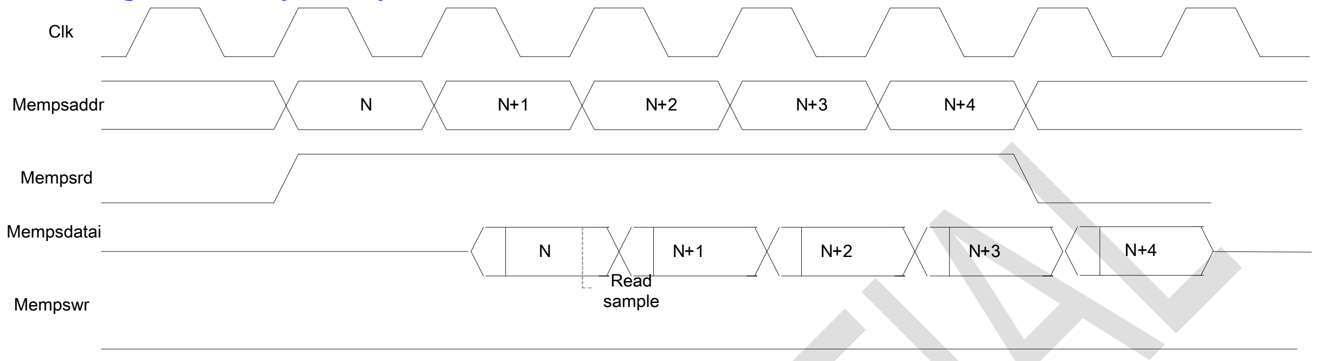
Set swdt (ien1.7).

And set swdt must be in 12 clock cycles after setting wdt. If this period has expired and swdt has not been set, wdt will be automatically reset. After refresh flag has been set, the watchdog timer will be reloaded with the contents of the wdtrel register and wdt bit is automatically reset.

6 Instruction Timing

6.1 Program memory bus cycle

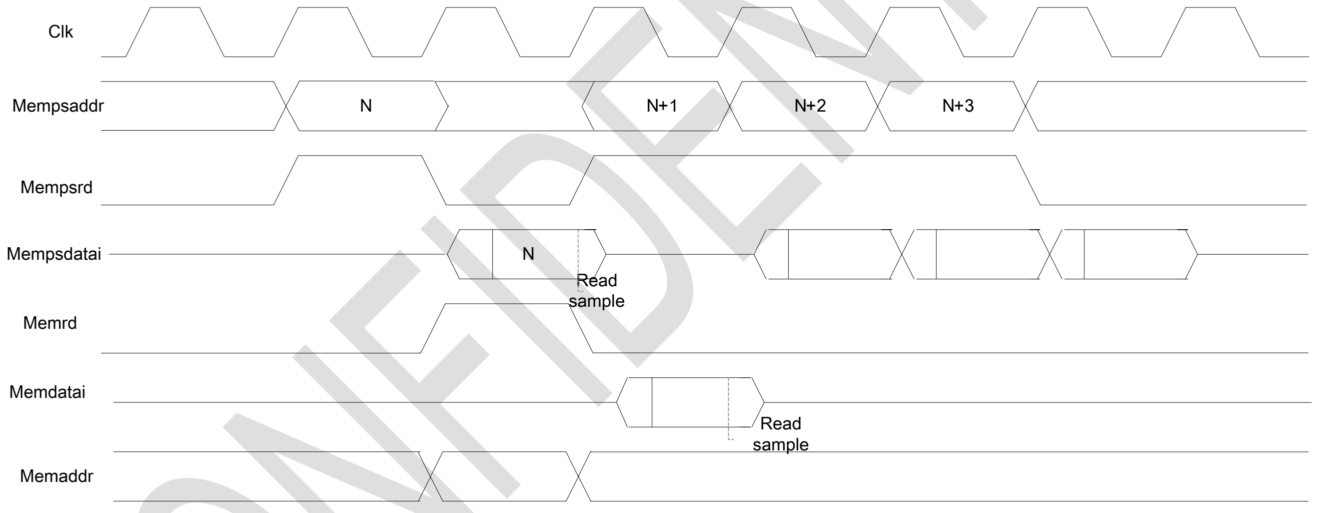
6.1.1 Program memory read cycle



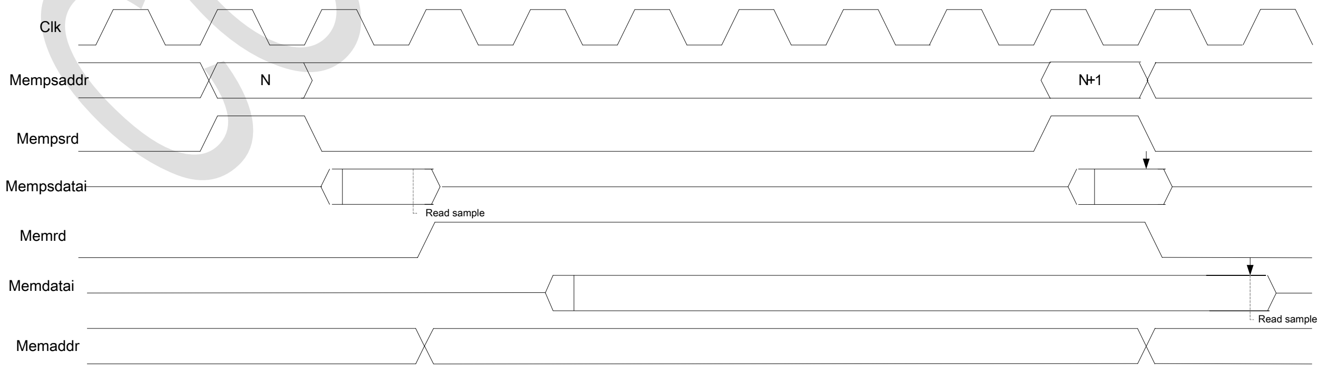
Program read without waitstates

6.2 External Data Memory bus cycle

6.2.1 External data memory read cycle

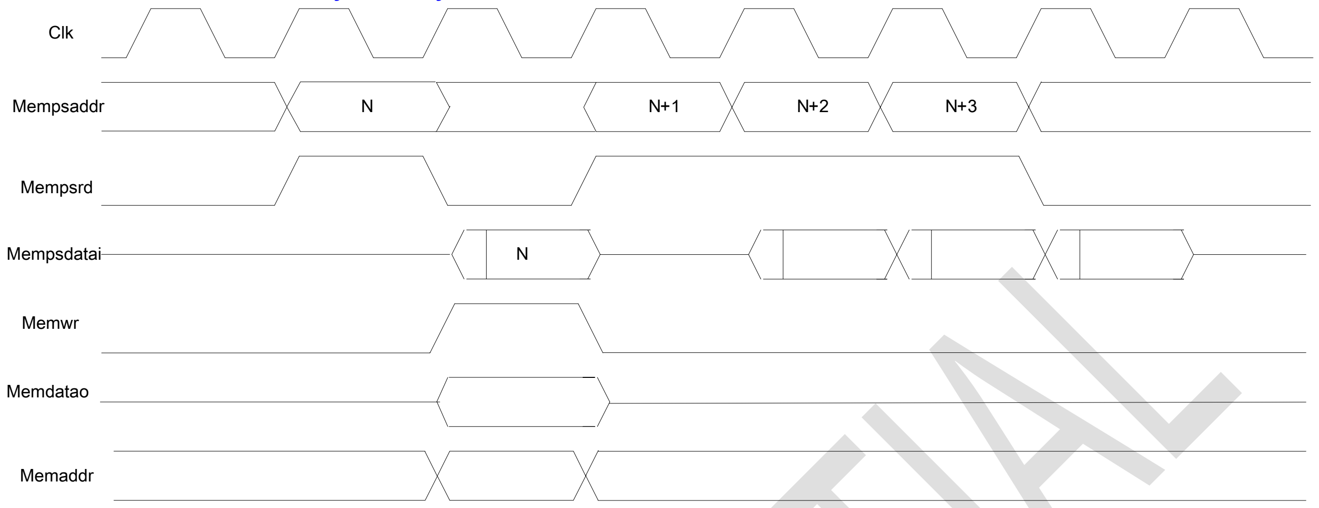


External data memory read without waitstates

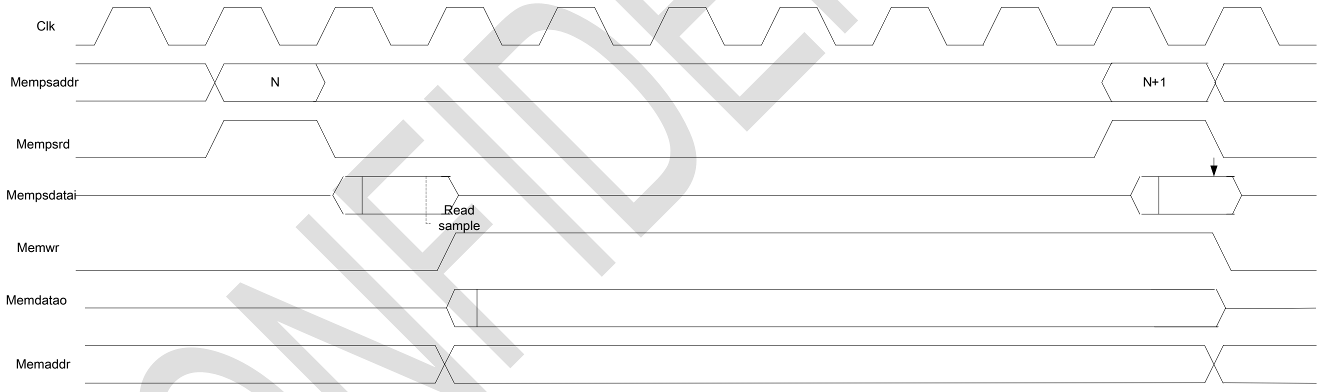


External data memory read with stretch 7

6.2.2 External data memory write cycle



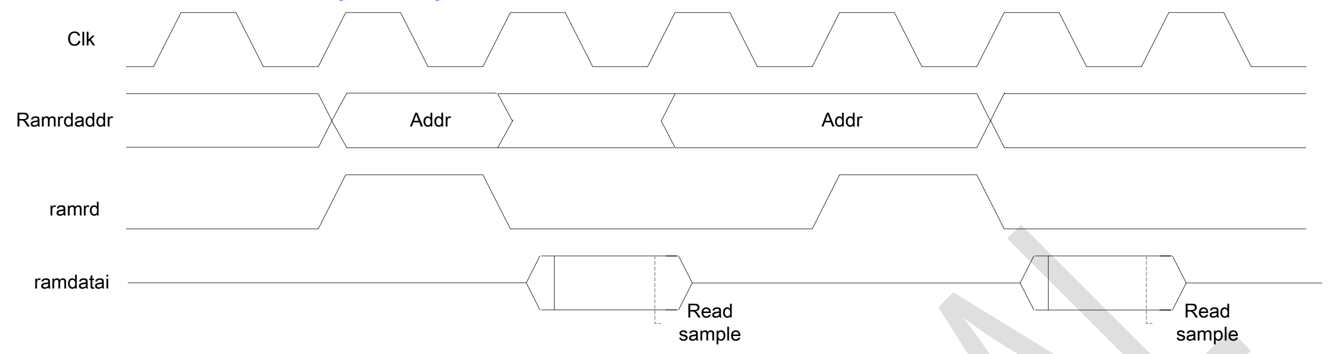
External data memory write without waitstates



External data memory write with stretch 7

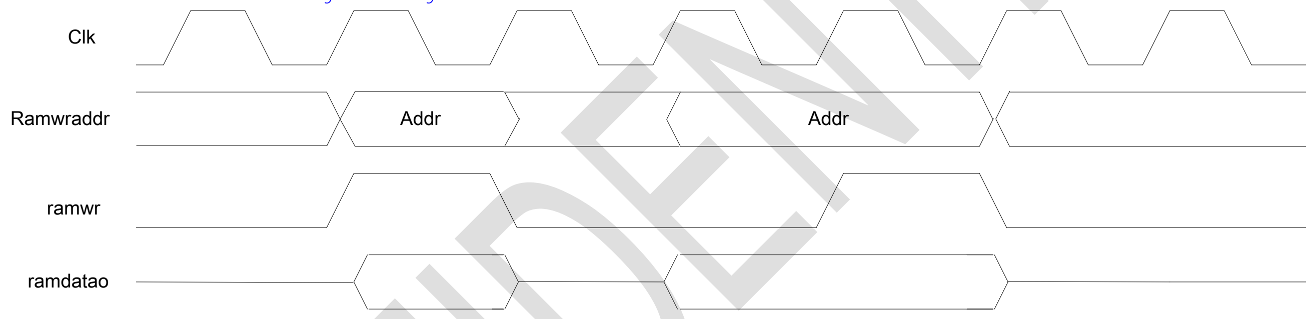
6.3 Internal Data Memory bus cycle

6.3.1 Internal data memory read cycle



Internal data memory read

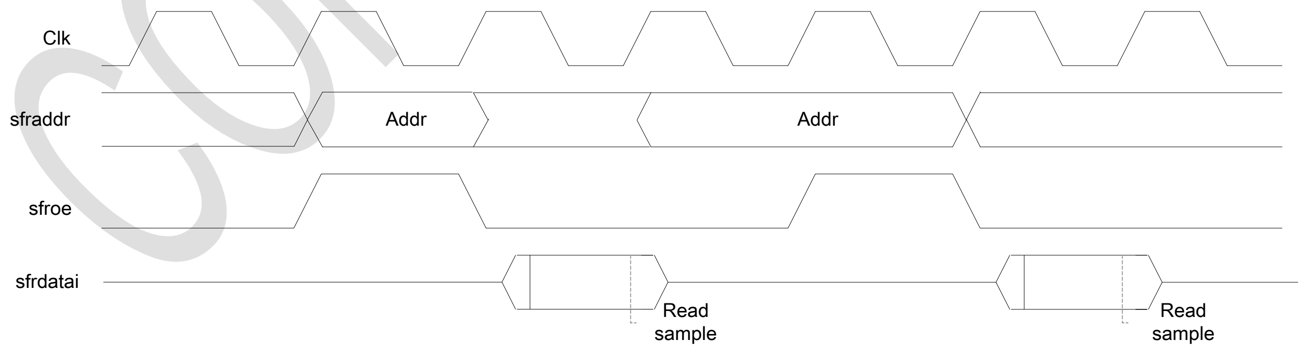
6.3.2 Internal data memory write cycle



Internal data memory write

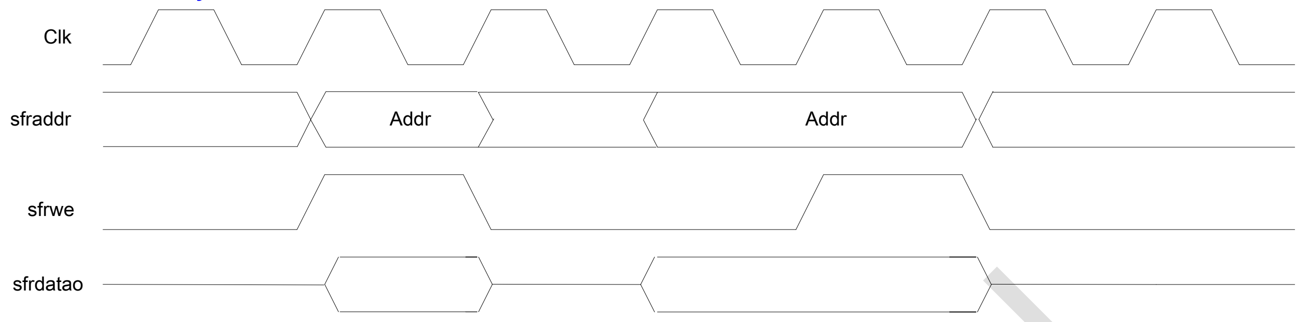
6.4 SFR bus cycle

6.4.1 SFR read cycle



SFR read

6.4.2 SFR write cycle



SFR write