```
/* ------------------------------------------------------------------------
--                                                                        --
--                            GNAT RAVENSCAR for NXT                      --
--                        Copyright (C) 2010, AdaCore                     --
--                                                                        --
------------------------------------------------------------------------ */

#ifndef AT91SAM7S_H
#  define AT91SAM7S_H

#  define CLOCK_FREQUENCY 48054850


// AIC peripheral ids
#  define AT91C_PERIPHERAL_ID_FIQ       0
#  define AT91C_PERIPHERAL_ID_SYSIRQ    1
#  define AT91C_PERIPHERAL_ID_PIOA      2
#  define AT91C_PERIPHERAL_ID_ADC       4
#  define AT91C_PERIPHERAL_ID_SPI       5
#  define AT91C_PERIPHERAL_ID_US0       6
#  define AT91C_PERIPHERAL_ID_US1       7
#  define AT91C_PERIPHERAL_ID_SSC       8
#  define AT91C_PERIPHERAL_ID_TWI       9
#  define AT91C_PERIPHERAL_ID_PWMC      10
#  define AT91C_PERIPHERAL_ID_UDP       11
#  define AT91C_PERIPHERAL_ID_TC0       12
#  define AT91C_PERIPHERAL_ID_TC1       13
#  define AT91C_PERIPHERAL_ID_TC2       14
#  define AT91C_PERIPHERAL_ID_IRQ0      30
#  define AT91C_PERIPHERAL_ID_IRQ1      31


typedef volatile unsigned int AT91_REG; // Hardware register definition

//
// ******************************************************************************
//               SOFTWARE API DEFINITION  FOR System Peripherals
//
// ******************************************************************************
typedef struct _AT91S_SYSC {
```

```
39  AT91_REG SYSC_AIC_SMR[32];    // Source Mode Register
40  AT91_REG SYSC_AIC_SVR[32];    // Source Vector Register
41  AT91_REG SYSC_AIC_IVR;    // IRQ Vector Register
42  AT91_REG SYSC_AIC_FVR;    // FIQ Vector Register
43  AT91_REG SYSC_AIC_ISR;    // Interrupt Status Register
44  AT91_REG SYSC_AIC_IPR;    // Interrupt Pending Register
45  AT91_REG SYSC_AIC_IMR;    // Interrupt Mask Register
46  AT91_REG SYSC_AIC_CISR;   // Core Interrupt Status Register
47  AT91_REG Reserved0[2];    //
48  AT91_REG SYSC_AIC_IECR;   // Interrupt Enable Command Register
49  AT91_REG SYSC_AIC_IDCR;   // Interrupt Disable Command Register
50  AT91_REG SYSC_AIC_ICCR;   // Interrupt Clear Command Register
51  AT91_REG SYSC_AIC_ISCR;   // Interrupt Set Command Register
52  AT91_REG SYSC_AIC_EOICR;  // End of Interrupt Command Register
53  AT91_REG SYSC_AIC_SPU;    // Spurious Vector Register
54  AT91_REG SYSC_AIC_DCR;    // Debug Control Register (Protect)
55  AT91_REG Reserved1[1];    //
56  AT91_REG SYSC_AIC_FFER;   // Fast Forcing Enable Register
57  AT91_REG SYSC_AIC_FFDR;   // Fast Forcing Disable Register
58  AT91_REG SYSC_AIC_FFSR;   // Fast Forcing Status Register
59  AT91_REG Reserved2[45];   //
60  AT91_REG SYSC_DBGU_CR;    // Control Register
61  AT91_REG SYSC_DBGU_MR;    // Mode Register
62  AT91_REG SYSC_DBGU_IER;   // Interrupt Enable Register
63  AT91_REG SYSC_DBGU_IDR;   // Interrupt Disable Register
64  AT91_REG SYSC_DBGU_IMR;   // Interrupt Mask Register
65  AT91_REG SYSC_DBGU_CSR;   // Channel Status Register
66  AT91_REG SYSC_DBGU_RHR;   // Receiver Holding Register
67  AT91_REG SYSC_DBGU_THR;   // Transmitter Holding Register
68  AT91_REG SYSC_DBGU_BRGR;  // Baud Rate Generator Register
69  AT91_REG Reserved3[7];    //
70  AT91_REG SYSC_DBGU_C1R;   // Chip ID1 Register
71  AT91_REG SYSC_DBGU_C2R;   // Chip ID2 Register
72  AT91_REG SYSC_DBGU_FNTR;  // Force NTRST Register
73  AT91_REG Reserved4[45];   //
74  AT91_REG SYSC_DBGU_RPR;   // Receive Pointer Register
75  AT91_REG SYSC_DBGU_RCR;   // Receive Counter Register
76  AT91_REG SYSC_DBGU_TPR;   // Transmit Pointer Register
77  AT91_REG SYSC_DBGU_TCR;   // Transmit Counter Register
78  AT91_REG SYSC_DBGU_RNPR;  // Receive Next Pointer Register
```

```
79  AT91_REG SYSC_DBGU_RNCR;  // Receive Next Counter Register
80  AT91_REG SYSC_DBGU_TNPR;  // Transmit Next Pointer Register
81  AT91_REG SYSC_DBGU_TNCR;  // Transmit Next Counter Register
82  AT91_REG SYSC_DBGU_PTCR;  // PDC Transfer Control Register
83  AT91_REG SYSC_DBGU_PTSR;  // PDC Transfer Status Register
84  AT91_REG Reserved5[54];   //
85  AT91_REG SYSC_PIOA_PER;   // PIO Enable Register
86  AT91_REG SYSC_PIOA_PDR;   // PIO Disable Register
87  AT91_REG SYSC_PIOA_PSR;   // PIO Status Register
88  AT91_REG Reserved6[1];    //
89  AT91_REG SYSC_PIOA_OER;   // Output Enable Register
90  AT91_REG SYSC_PIOA_ODR;   // Output Disable Register
91  AT91_REG SYSC_PIOA_OSR;   // Output Status Register
92  AT91_REG Reserved7[1];    //
93  AT91_REG SYSC_PIOA_IFER;  // Input Filter Enable Register
94  AT91_REG SYSC_PIOA_IFDR;  // Input Filter Disable Register
95  AT91_REG SYSC_PIOA_IFSR;  // Input Filter Status Register
96  AT91_REG Reserved8[1];    //
97  AT91_REG SYSC_PIOA_SODR;  // Set Output Data Register
98  AT91_REG SYSC_PIOA_CODR;  // Clear Output Data Register
99  AT91_REG SYSC_PIOA_ODSR;  // Output Data Status Register
100 AT91_REG SYSC_PIOA_PDSR;  // Pin Data Status Register
101 AT91_REG SYSC_PIOA_IER;   // Interrupt Enable Register
102 AT91_REG SYSC_PIOA_IDR;   // Interrupt Disable Register
103 AT91_REG SYSC_PIOA_IMR;   // Interrupt Mask Register
104 AT91_REG SYSC_PIOA_ISR;   // Interrupt Status Register
105 AT91_REG SYSC_PIOA_MDER;  // Multi-driver Enable Register
106 AT91_REG SYSC_PIOA_MDDR;  // Multi-driver Disable Register
107 AT91_REG SYSC_PIOA_MDSR;  // Multi-driver Status Register
108 AT91_REG Reserved9[1];    //
109 AT91_REG SYSC_PIOA_PPUDR; // Pull-up Disable Register
110 AT91_REG SYSC_PIOA_PPUER; // Pull-up Enable Register
111 AT91_REG SYSC_PIOA_PPUSR; // Pad Pull-up Status Register
112 AT91_REG Reserved10[1];   //
113 AT91_REG SYSC_PIOA_ASR;   // Select A Register
114 AT91_REG SYSC_PIOA_BSR;   // Select B Register
115 AT91_REG SYSC_PIOA_ABSR;  // AB Select Status Register
116 AT91_REG Reserved11[9];   //
117 AT91_REG SYSC_PIOA_OWER;  // Output Write Enable Register
118 AT91_REG SYSC_PIOA_OWDR;  // Output Write Disable Register
```

```
119   AT91_REG SYSC_PIOA_OWSR;  // Output Write Status Register
120   AT91_REG Reserved12[469]; //
121   AT91_REG SYSC_PMC_SCER;   // System Clock Enable Register
122   AT91_REG SYSC_PMC_SCDR;   // System Clock Disable Register
123   AT91_REG SYSC_PMC_SCSR;   // System Clock Status Register
124   AT91_REG Reserved13[1];   //
125   AT91_REG SYSC_PMC_PCER;   // Peripheral Clock Enable Register
126   AT91_REG SYSC_PMC_PCDR;   // Peripheral Clock Disable Register
127   AT91_REG SYSC_PMC_PCSR;   // Peripheral Clock Status Register
128   AT91_REG Reserved14[1];   //
129   AT91_REG SYSC_PMC_MOR;    // Main Oscillator Register
130   AT91_REG SYSC_PMC_MCFR;   // Main Clock  Frequency Register
131   AT91_REG Reserved15[1];   //
132   AT91_REG SYSC_PMC_PLLR;   // PLL Register
133   AT91_REG SYSC_PMC_MCKR;   // Master Clock Register
134   AT91_REG Reserved16[3];   //
135   AT91_REG SYSC_PMC_PCKR[8];    // Programmable Clock Register
136   AT91_REG SYSC_PMC_IER;    // Interrupt Enable Register
137   AT91_REG SYSC_PMC_IDR;    // Interrupt Disable Register
138   AT91_REG SYSC_PMC_SR;     // Status Register
139   AT91_REG SYSC_PMC_IMR;    // Interrupt Mask Register
140   AT91_REG Reserved17[36];  //
141   AT91_REG SYSC_RSTC_RCR;   // Reset Control Register
142   AT91_REG SYSC_RSTC_RSR;   // Reset Status Register
143   AT91_REG SYSC_RSTC_RMR;   // Reset Mode Register
144   AT91_REG Reserved18[5];   //
145   AT91_REG SYSC_RTTC_RTMR;  // Real-time Mode Register
146   AT91_REG SYSC_RTTC_RTAR;  // Real-time Alarm Register
147   AT91_REG SYSC_RTTC_RTVR;  // Real-time Value Register
148   AT91_REG SYSC_RTTC_RTSR;  // Real-time Status Register
149   AT91_REG SYSC_PITC_PIMR;  // Period Interval Mode Register
150   AT91_REG SYSC_PITC_PISR;  // Period Interval Status Register
151   AT91_REG SYSC_PITC_PIVR;  // Period Interval Value Register
152   AT91_REG SYSC_PITC_PIIR;  // Period Interval Image Register
153   AT91_REG SYSC_WDTC_WDCR;  // Watchdog Control Register
154   AT91_REG SYSC_WDTC_WDMR;  // Watchdog Mode Register
155   AT91_REG SYSC_WDTC_WDSR;  // Watchdog Status Register
156   AT91_REG Reserved19[5];   //
157   AT91_REG SYSC_SYSC_VREG;  // Voltage Regulator Mode Register
158 } AT91S_SYSC, *AT91PS_SYSC;
```

```
159
160 // -------- VREG : (SYSC Offset: 0xd60) Voltage Regulator Mode Register
    --------
161 #  define AT91C_SYSC_PSTDBY     ((unsigned int) 0x1 <<  0)  // (SYSC) Voltage
    Regulator Power Mode
162
163 //
    ******************************************************************************
164 //               SOFTWARE API DEFINITION  FOR Advanced Interrupt Controller
165 //
    ******************************************************************************
166 typedef struct _AT91S_AIC {
167   AT91_REG AIC_SMR[32];     // Source Mode Register
168   AT91_REG AIC_SVR[32];     // Source Vector Register
169   AT91_REG AIC_IVR;     // IRQ Vector Register
170   AT91_REG AIC_FVR;     // FIQ Vector Register
171   AT91_REG AIC_ISR;     // Interrupt Status Register
172   AT91_REG AIC_IPR;     // Interrupt Pending Register
173   AT91_REG AIC_IMR;     // Interrupt Mask Register
174   AT91_REG AIC_CISR;        // Core Interrupt Status Register
175   AT91_REG Reserved0[2];    //
176   AT91_REG AIC_IECR;        // Interrupt Enable Command Register
177   AT91_REG AIC_IDCR;        // Interrupt Disable Command Register
178   AT91_REG AIC_ICCR;        // Interrupt Clear Command Register
179   AT91_REG AIC_ISCR;        // Interrupt Set Command Register
180   AT91_REG AIC_EOICR;       // End of Interrupt Command Register
181   AT91_REG AIC_SPU;     // Spurious Vector Register
182   AT91_REG AIC_DCR;     // Debug Control Register (Protect)
183   AT91_REG Reserved1[1];    //
184   AT91_REG AIC_FFER;        // Fast Forcing Enable Register
185   AT91_REG AIC_FFDR;        // Fast Forcing Disable Register
186   AT91_REG AIC_FFSR;        // Fast Forcing Status Register
187 } AT91S_AIC, *AT91PS_AIC;
188
189 // -------- AIC_SMR : (AIC Offset: 0x0) Control Register --------
190 #  define AT91C_AIC_PRIOR       ((unsigned int) 0x7 <<  0)  // (AIC) Priority
    Level
191 #  define   AT91C_AIC_PRIOR_LOWEST              ((unsigned int) 0x0)   //
    (AIC) Lowest priority level
192 #  define   AT91C_AIC_PRIOR_HIGHEST             ((unsigned int) 0x7)   //
```

```
192… (AIC) Highest priority level
193 #  define AT91C_AIC_SRCTYPE     ((unsigned int) 0x3 <<  5)  // (AIC) Interrupt
… Source Type
194 #  define   AT91C_AIC_SRCTYPE_INT_LEVEL_SENSITIVE  ((unsigned int) 0x0 <<  5)
… // (AIC) Internal Sources Code Label Level Sensitive
195 #  define   AT91C_AIC_SRCTYPE_INT_EDGE_TRIGGERED   ((unsigned int) 0x1 <<  5)
… // (AIC) Internal Sources Code Label Edge triggered
196 #  define   AT91C_AIC_SRCTYPE_EXT_HIGH_LEVEL       ((unsigned int) 0x2 <<  5)
… // (AIC) External Sources Code Label High-level Sensitive
197 #  define   AT91C_AIC_SRCTYPE_EXT_POSITIVE_EDGE    ((unsigned int) 0x3 <<  5)
… // (AIC) External Sources Code Label Positive Edge triggered
198 // -------- AIC_CISR : (AIC Offset: 0x114) AIC Core Interrupt Status Register
… --------
199 #  define AT91C_AIC_NFIQ        ((unsigned int) 0x1 <<  0)  // (AIC) NFIQ
… Status
200 #  define AT91C_AIC_NIRQ        ((unsigned int) 0x1 <<  1)  // (AIC) NIRQ
… Status
201 // -------- AIC_DCR : (AIC Offset: 0x138) AIC Debug Control Register (Protect)
… --------
202 #  define AT91C_AIC_DCR_PROT    ((unsigned int) 0x1 <<  0)  // (AIC) Protection
… Mode
203 #  define AT91C_AIC_DCR_GMSK    ((unsigned int) 0x1 <<  1)  // (AIC) General
… Mask
204
205 //
… ////////////////////////////////////////////////////////////////////////////
206 //               SOFTWARE API DEFINITION  FOR Debug Unit
207 //
… ////////////////////////////////////////////////////////////////////////////
208 typedef struct _AT91S_DBGU {
209   AT91_REG DBGU_CR;     // Control Register
210   AT91_REG DBGU_MR;     // Mode Register
211   AT91_REG DBGU_IER;        // Interrupt Enable Register
212   AT91_REG DBGU_IDR;        // Interrupt Disable Register
213   AT91_REG DBGU_IMR;        // Interrupt Mask Register
214   AT91_REG DBGU_CSR;        // Channel Status Register
215   AT91_REG DBGU_RHR;        // Receiver Holding Register
216   AT91_REG DBGU_THR;        // Transmitter Holding Register
217   AT91_REG DBGU_BRGR;       // Baud Rate Generator Register
218   AT91_REG Reserved0[7];    //
```

```
219   AT91_REG DBGU_C1R;        // Chip ID1 Register
220   AT91_REG DBGU_C2R;        // Chip ID2 Register
221   AT91_REG DBGU_FNTR;       // Force NTRST Register
222   AT91_REG Reserved1[45];   //
223   AT91_REG DBGU_RPR;        // Receive Pointer Register
224   AT91_REG DBGU_RCR;        // Receive Counter Register
225   AT91_REG DBGU_TPR;        // Transmit Pointer Register
226   AT91_REG DBGU_TCR;        // Transmit Counter Register
227   AT91_REG DBGU_RNPR;       // Receive Next Pointer Register
228   AT91_REG DBGU_RNCR;       // Receive Next Counter Register
229   AT91_REG DBGU_TNPR;       // Transmit Next Pointer Register
230   AT91_REG DBGU_TNCR;       // Transmit Next Counter Register
231   AT91_REG DBGU_PTCR;       // PDC Transfer Control Register
232   AT91_REG DBGU_PTSR;       // PDC Transfer Status Register
233 } AT91S_DBGU, *AT91PS_DBGU;
234 
235 // -------- DBGU_CR : (DBGU Offset: 0x0) Debug Unit Control Register --------
236 #  define AT91C_US_RSTRX        ((unsigned int) 0x1 <<  2)  // (DBGU) Reset
    … Receiver
237 #  define AT91C_US_RSTTX        ((unsigned int) 0x1 <<  3)  // (DBGU) Reset
    … Transmitter
238 #  define AT91C_US_RXEN         ((unsigned int) 0x1 <<  4)  // (DBGU) Receiver
    … Enable
239 #  define AT91C_US_RXDIS        ((unsigned int) 0x1 <<  5)  // (DBGU) Receiver
    … Disable
240 #  define AT91C_US_TXEN         ((unsigned int) 0x1 <<  6)  // (DBGU)
    … Transmitter Enable
241 #  define AT91C_US_TXDIS        ((unsigned int) 0x1 <<  7)  // (DBGU)
    … Transmitter Disable
242 // -------- DBGU_MR : (DBGU Offset: 0x4) Debug Unit Mode Register --------
243 #  define AT91C_US_PAR          ((unsigned int) 0x7 <<  9)  // (DBGU) Parity
    … type
244 #  define   AT91C_US_PAR_EVEN                ((unsigned int) 0x0 <<  9)    //
    … (DBGU) Even Parity
245 #  define   AT91C_US_PAR_ODD                 ((unsigned int) 0x1 <<  9)    //
    … (DBGU) Odd Parity
246 #  define   AT91C_US_PAR_SPACE               ((unsigned int) 0x2 <<  9)    //
    … (DBGU) Parity forced to 0 (Space)
247 #  define   AT91C_US_PAR_MARK                ((unsigned int) 0x3 <<  9)    //
    … (DBGU) Parity forced to 1 (Mark)
```

```
248 #  define   AT91C_US_PAR_NONE                 ((unsigned int) 0x4 <<  9)    //
… (DBGU) No Parity
249 #  define   AT91C_US_PAR_MULTI_DROP           ((unsigned int) 0x6 <<  9)    //
… (DBGU) Multi-drop mode
250 #  define AT91C_US_CHMODE       ((unsigned int) 0x3 << 14)  // (DBGU) Channel
… Mode
251 #  define   AT91C_US_CHMODE_NORMAL                ((unsigned int) 0x0 << 14) //
… (DBGU) Normal Mode: The USART channel operates as an RX/TX USART.
252 #  define   AT91C_US_CHMODE_AUTO                  ((unsigned int) 0x1 << 14) //
… (DBGU) Automatic Echo: Receiver Data Input is connected to the TXD pin.
253 #  define   AT91C_US_CHMODE_LOCAL                 ((unsigned int) 0x2 << 14) //
… (DBGU) Local Loopback: Transmitter Output Signal is connected to Receiver Input
… Signal.
254 #  define   AT91C_US_CHMODE_REMOTE                ((unsigned int) 0x3 << 14) //
… (DBGU) Remote Loopback: RXD pin is internally connected to TXD pin.
255 // -------- DBGU_IER : (DBGU Offset: 0x8) Debug Unit Interrupt Enable Register
… --------
256 #  define AT91C_US_RXRDY        ((unsigned int) 0x1 <<  0)  // (DBGU) RXRDY
… Interrupt
257 #  define AT91C_US_TXRDY        ((unsigned int) 0x1 <<  1)  // (DBGU) TXRDY
… Interrupt
258 #  define AT91C_US_ENDRX        ((unsigned int) 0x1 <<  3)  // (DBGU) End of
… Receive Transfer Interrupt
259 #  define AT91C_US_ENDTX        ((unsigned int) 0x1 <<  4)  // (DBGU) End of
… Transmit Interrupt
260 #  define AT91C_US_OVRE         ((unsigned int) 0x1 <<  5)  // (DBGU) Overrun
… Interrupt
261 #  define AT91C_US_FRAME        ((unsigned int) 0x1 <<  6)  // (DBGU) Framing
… Error Interrupt
262 #  define AT91C_US_PARE         ((unsigned int) 0x1 <<  7)  // (DBGU) Parity
… Error Interrupt
263 #  define AT91C_US_TXEMPTY      ((unsigned int) 0x1 <<  9)  // (DBGU) TXEMPTY
… Interrupt
264 #  define AT91C_US_TXBUFE       ((unsigned int) 0x1 << 11)  // (DBGU) TXBUFE
… Interrupt
265 #  define AT91C_US_RXBUFF       ((unsigned int) 0x1 << 12)  // (DBGU) RXBUFF
… Interrupt
266 #  define AT91C_US_COMM_TX      ((unsigned int) 0x1 << 30)  // (DBGU) COMM_TX
… Interrupt
267 #  define AT91C_US_COMM_RX      ((unsigned int) 0x1 << 31)  // (DBGU) COMM_RX
```

```
Interrupt
// -------- DBGU_IDR : (DBGU Offset: 0xc) Debug Unit Interrupt Disable Register --------
// -------- DBGU_IMR : (DBGU Offset: 0x10) Debug Unit Interrupt Mask Register --------
// -------- DBGU_CSR : (DBGU Offset: 0x14) Debug Unit Channel Status Register --------
// -------- DBGU_FNTR : (DBGU Offset: 0x48) Debug Unit FORCE_NTRST Register --------
#  define AT91C_US_FORCE_NTRST  ((unsigned int) 0x1 <<  0)  // (DBGU) Force NTRST in JTAG

//
*******************************************************************************
//              SOFTWARE API DEFINITION  FOR Peripheral Data Controller
//
*******************************************************************************
typedef struct _AT91S_PDC {
  AT91_REG PDC_RPR;     // Receive Pointer Register
  AT91_REG PDC_RCR;     // Receive Counter Register
  AT91_REG PDC_TPR;     // Transmit Pointer Register
  AT91_REG PDC_TCR;     // Transmit Counter Register
  AT91_REG PDC_RNPR;        // Receive Next Pointer Register
  AT91_REG PDC_RNCR;        // Receive Next Counter Register
  AT91_REG PDC_TNPR;        // Transmit Next Pointer Register
  AT91_REG PDC_TNCR;        // Transmit Next Counter Register
  AT91_REG PDC_PTCR;        // PDC Transfer Control Register
  AT91_REG PDC_PTSR;        // PDC Transfer Status Register
} AT91S_PDC, *AT91PS_PDC;

// -------- PDC_PTCR : (PDC Offset: 0x20) PDC Transfer Control Register --------
#  define AT91C_PDC_RXTEN       ((unsigned int) 0x1 <<  0)  // (PDC) Receiver Transfer Enable
#  define AT91C_PDC_RXTDIS      ((unsigned int) 0x1 <<  1)  // (PDC) Receiver Transfer Disable
#  define AT91C_PDC_TXTEN       ((unsigned int) 0x1 <<  8)  // (PDC) Transmitter Transfer Enable
#  define AT91C_PDC_TXTDIS      ((unsigned int) 0x1 <<  9)  // (PDC) Transmitter Transfer Disable
```

```
295 // -------- PDC_PTSR : (PDC Offset: 0x24) PDC Transfer Status Register --------
296
297 //
    ...
298 //               SOFTWARE API DEFINITION  FOR Parallel Input Output Controler
299 //
    ...
300 typedef struct _AT91S_PIO {
301   AT91_REG PIO_PER;     // PIO Enable Register
302   AT91_REG PIO_PDR;     // PIO Disable Register
303   AT91_REG PIO_PSR;     // PIO Status Register
304   AT91_REG Reserved0[1];    //
305   AT91_REG PIO_OER;     // Output Enable Register
306   AT91_REG PIO_ODR;     // Output Disable Register
307   AT91_REG PIO_OSR;     // Output Status Register
308   AT91_REG Reserved1[1];    //
309   AT91_REG PIO_IFER;        // Input Filter Enable Register
310   AT91_REG PIO_IFDR;        // Input Filter Disable Register
311   AT91_REG PIO_IFSR;        // Input Filter Status Register
312   AT91_REG Reserved2[1];    //
313   AT91_REG PIO_SODR;        // Set Output Data Register
314   AT91_REG PIO_CODR;        // Clear Output Data Register
315   AT91_REG PIO_ODSR;        // Output Data Status Register
316   AT91_REG PIO_PDSR;        // Pin Data Status Register
317   AT91_REG PIO_IER;     // Interrupt Enable Register
318   AT91_REG PIO_IDR;     // Interrupt Disable Register
319   AT91_REG PIO_IMR;     // Interrupt Mask Register
320   AT91_REG PIO_ISR;     // Interrupt Status Register
321   AT91_REG PIO_MDER;        // Multi-driver Enable Register
322   AT91_REG PIO_MDDR;        // Multi-driver Disable Register
323   AT91_REG PIO_MDSR;        // Multi-driver Status Register
324   AT91_REG Reserved3[1];    //
325   AT91_REG PIO_PPUDR;       // Pull-up Disable Register
326   AT91_REG PIO_PPUER;       // Pull-up Enable Register
327   AT91_REG PIO_PPUSR;       // Pad Pull-up Status Register
328   AT91_REG Reserved4[1];    //
329   AT91_REG PIO_ASR;     // Select A Register
330   AT91_REG PIO_BSR;     // Select B Register
331   AT91_REG PIO_ABSR;        // AB Select Status Register
332   AT91_REG Reserved5[9];    //
```

```
333   AT91_REG PIO_OWER;        // Output Write Enable Register
334   AT91_REG PIO_OWDR;        // Output Write Disable Register
335   AT91_REG PIO_OWSR;        // Output Write Status Register
336 } AT91S_PIO, *AT91PS_PIO;
337
338
339 //
    ********************************************************************************
340 //               SOFTWARE API DEFINITION  FOR Clock Generator Controler
341 //
    ********************************************************************************
342 typedef struct _AT91S_CKGR {
343   AT91_REG CKGR_MOR;        // Main Oscillator Register
344   AT91_REG CKGR_MCFR;       // Main Clock  Frequency Register
345   AT91_REG Reserved0[1];    //
346   AT91_REG CKGR_PLLR;       // PLL Register
347 } AT91S_CKGR, *AT91PS_CKGR;
348
349 // -------- CKGR_MOR : (CKGR Offset: 0x0) Main Oscillator Register --------
350 #  define AT91C_CKGR_MOSCEN     ((unsigned int) 0x1 <<  0)  // (CKGR) Main
    Oscillator Enable
351 #  define AT91C_CKGR_OSCBYPASS  ((unsigned int) 0x1 <<  1)  // (CKGR) Main
    Oscillator Bypass
352 #  define AT91C_CKGR_OSCOUNT    ((unsigned int) 0xFF <<  8) // (CKGR) Main
    Oscillator Start-up Time
353 // -------- CKGR_MCFR : (CKGR Offset: 0x4) Main Clock Frequency Register
    --------
354 #  define AT91C_CKGR_MAINF      ((unsigned int) 0xFFFF <<  0)   // (CKGR) Main
    Clock Frequency
355 #  define AT91C_CKGR_MAINRDY    ((unsigned int) 0x1 << 16)  // (CKGR) Main
    Clock Ready
356 // -------- CKGR_PLLR : (CKGR Offset: 0xc) PLL B Register --------
357 #  define AT91C_CKGR_DIV        ((unsigned int) 0xFF <<  0) // (CKGR) Divider
    Selected
358 #  define   AT91C_CKGR_DIV_0                    ((unsigned int) 0x0)    //
    (CKGR) Divider output is 0
359 #  define   AT91C_CKGR_DIV_BYPASS               ((unsigned int) 0x1)    //
    (CKGR) Divider is bypassed
360 #  define AT91C_CKGR_PLLCOUNT   ((unsigned int) 0x3F <<  8) // (CKGR) PLL
    Counter
```

```
361 #  define AT91C_CKGR_OUT        ((unsigned int) 0x3 << 14)  // (CKGR) PLL
    … Output Frequency Range
362 #  define   AT91C_CKGR_OUT_0                    ((unsigned int) 0x0 << 14)  //
    … (CKGR) Please refer to the PLL datasheet
363 #  define   AT91C_CKGR_OUT_1                    ((unsigned int) 0x1 << 14)  //
    … (CKGR) Please refer to the PLL datasheet
364 #  define   AT91C_CKGR_OUT_2                    ((unsigned int) 0x2 << 14)  //
    … (CKGR) Please refer to the PLL datasheet
365 #  define   AT91C_CKGR_OUT_3                    ((unsigned int) 0x3 << 14)  //
    … (CKGR) Please refer to the PLL datasheet
366 #  define AT91C_CKGR_MUL        ((unsigned int) 0x7FF << 16)    // (CKGR) PLL
    … Multiplier
367 #  define AT91C_CKGR_USBDIV     ((unsigned int) 0x3 << 28)  // (CKGR) Divider
    … for USB Clocks
368 #  define   AT91C_CKGR_USBDIV_0                     ((unsigned int) 0x0 << 28)
    … // (CKGR) Divider output is PLL clock output
369 #  define   AT91C_CKGR_USBDIV_1                     ((unsigned int) 0x1 << 28)
    … // (CKGR) Divider output is PLL clock output divided by 2
370 #  define   AT91C_CKGR_USBDIV_2                     ((unsigned int) 0x2 << 28)
    … // (CKGR) Divider output is PLL clock output divided by 4
371
372 //
    … ********************************************************************************
373 //               SOFTWARE API DEFINITION  FOR Power Management Controler
374 //
    … ********************************************************************************
375 typedef struct _AT91S_PMC {
376   AT91_REG PMC_SCER;        // System Clock Enable Register
377   AT91_REG PMC_SCDR;        // System Clock Disable Register
378   AT91_REG PMC_SCSR;        // System Clock Status Register
379   AT91_REG Reserved0[1];    //
380   AT91_REG PMC_PCER;        // Peripheral Clock Enable Register
381   AT91_REG PMC_PCDR;        // Peripheral Clock Disable Register
382   AT91_REG PMC_PCSR;        // Peripheral Clock Status Register
383   AT91_REG Reserved1[1];    //
384   AT91_REG PMC_MOR;     // Main Oscillator Register
385   AT91_REG PMC_MCFR;        // Main Clock  Frequency Register
386   AT91_REG Reserved2[1];    //
387   AT91_REG PMC_PLLR;        // PLL Register
388   AT91_REG PMC_MCKR;        // Master Clock Register
```

```
389   AT91_REG Reserved3[3];    //
390   AT91_REG PMC_PCKR[8];     // Programmable Clock Register
391   AT91_REG PMC_IER;     // Interrupt Enable Register
392   AT91_REG PMC_IDR;     // Interrupt Disable Register
393   AT91_REG PMC_SR;      // Status Register
394   AT91_REG PMC_IMR;     // Interrupt Mask Register
395 } AT91S_PMC, *AT91PS_PMC;
396
397 // -------- PMC_SCER : (PMC Offset: 0x0) System Clock Enable Register --------
398 #  define AT91C_PMC_PCK         ((unsigned int) 0x1 <<  0)  // (PMC) Processor
    … Clock
399 #  define AT91C_PMC_UDP         ((unsigned int) 0x1 <<  7)  // (PMC) USB Device
    … Port Clock
400 #  define AT91C_PMC_PCK0        ((unsigned int) 0x1 <<  8)  // (PMC)
    … Programmable Clock Output
401 #  define AT91C_PMC_PCK1        ((unsigned int) 0x1 <<  9)  // (PMC)
    … Programmable Clock Output
402 #  define AT91C_PMC_PCK2        ((unsigned int) 0x1 << 10)  // (PMC)
    … Programmable Clock Output
403 #  define AT91C_PMC_PCK3        ((unsigned int) 0x1 << 11)  // (PMC)
    … Programmable Clock Output
404 // -------- PMC_SCDR : (PMC Offset: 0x4) System Clock Disable Register --------
405 // -------- PMC_SCSR : (PMC Offset: 0x8) System Clock Status Register --------
406 // -------- CKGR_MOR : (PMC Offset: 0x20) Main Oscillator Register --------
407 // -------- CKGR_MCFR : (PMC Offset: 0x24) Main Clock Frequency Register
    … --------
408 // -------- CKGR_PLLR : (PMC Offset: 0x2c) PLL B Register --------
409 // -------- PMC_MCKR : (PMC Offset: 0x30) Master Clock Register --------
410 #  define AT91C_PMC_CSS         ((unsigned int) 0x3 <<  0)  // (PMC)
    … Programmable Clock Selection
411 #  define   AT91C_PMC_CSS_SLOW_CLK             ((unsigned int) 0x0) // (PMC)
    … Slow Clock is selected
412 #  define   AT91C_PMC_CSS_MAIN_CLK             ((unsigned int) 0x1) // (PMC)
    … Main Clock is selected
413 #  define   AT91C_PMC_CSS_PLL_CLK              ((unsigned int) 0x3) // (PMC)
    … Clock from PLL is selected
414 #  define AT91C_PMC_PRES        ((unsigned int) 0x7 <<  2)  // (PMC)
    … Programmable Clock Prescaler
415 #  define   AT91C_PMC_PRES_CLK                  ((unsigned int) 0x0 <<  2)  //
    … (PMC) Selected clock
```

```
416 #  define   AT91C_PMC_PRES_CLK_2              ((unsigned int) 0x1 <<  2)  //
… (PMC) Selected clock divided by 2
417 #  define   AT91C_PMC_PRES_CLK_4              ((unsigned int) 0x2 <<  2)  //
… (PMC) Selected clock divided by 4
418 #  define   AT91C_PMC_PRES_CLK_8              ((unsigned int) 0x3 <<  2)  //
… (PMC) Selected clock divided by 8
419 #  define   AT91C_PMC_PRES_CLK_16             ((unsigned int) 0x4 <<  2)  //
… (PMC) Selected clock divided by 16
420 #  define   AT91C_PMC_PRES_CLK_32             ((unsigned int) 0x5 <<  2)  //
… (PMC) Selected clock divided by 32
421 #  define   AT91C_PMC_PRES_CLK_64             ((unsigned int) 0x6 <<  2)  //
… (PMC) Selected clock divided by 64
422 // -------- PMC_PCKR : (PMC Offset: 0x40) Programmable Clock Register --------
423 // -------- PMC_IER : (PMC Offset: 0x60) PMC Interrupt Enable Register --------
424 #  define AT91C_PMC_MOSCS       ((unsigned int) 0x1 <<  0)  // (PMC) MOSC
… Status/Enable/Disable/Mask
425 #  define AT91C_PMC_LOCK        ((unsigned int) 0x1 <<  2)  // (PMC) PLL
… Status/Enable/Disable/Mask
426 #  define AT91C_PMC_MCKRDY      ((unsigned int) 0x1 <<  3)  // (PMC) MCK_RDY
… Status/Enable/Disable/Mask
427 #  define AT91C_PMC_PCK0RDY     ((unsigned int) 0x1 <<  8)  // (PMC) PCK0_RDY
… Status/Enable/Disable/Mask
428 #  define AT91C_PMC_PCK1RDY     ((unsigned int) 0x1 <<  9)  // (PMC) PCK1_RDY
… Status/Enable/Disable/Mask
429 #  define AT91C_PMC_PCK2RDY     ((unsigned int) 0x1 << 10)  // (PMC) PCK2_RDY
… Status/Enable/Disable/Mask
430 #  define AT91C_PMC_PCK3RDY     ((unsigned int) 0x1 << 11)  // (PMC) PCK3_RDY
… Status/Enable/Disable/Mask
431 // -------- PMC_IDR : (PMC Offset: 0x64) PMC Interrupt Disable Register
… --------
432 // -------- PMC_SR : (PMC Offset: 0x68) PMC Status Register --------
433 // -------- PMC_IMR : (PMC Offset: 0x6c) PMC Interrupt Mask Register --------
434
435 //
… ////////////////////////////////////////////////////////////////////////////
436 //               SOFTWARE API DEFINITION  FOR Reset Controller Interface
437 //
… ////////////////////////////////////////////////////////////////////////////
438 typedef struct _AT91S_RSTC {
439   AT91_REG RSTC_RCR;        // Reset Control Register
```

```
440   AT91_REG RSTC_RSR;        // Reset Status Register
441   AT91_REG RSTC_RMR;        // Reset Mode Register
442 } AT91S_RSTC, *AT91PS_RSTC;
443
444 // -------- SYSC_RCR : (RSTC Offset: 0x0) Reset Control Register --------
445 #  define AT91C_SYSC_PROCRST    ((unsigned int) 0x1 <<  0)  // (RSTC) Processor
    … Reset
446 #  define AT91C_SYSC_ICERST     ((unsigned int) 0x1 <<  1)  // (RSTC) ICE
    … Interface Reset
447 #  define AT91C_SYSC_PERRST     ((unsigned int) 0x1 <<  2)  // (RSTC)
    … Peripheral Reset
448 #  define AT91C_SYSC_EXTRST     ((unsigned int) 0x1 <<  3)  // (RSTC) External
    … Reset
449 #  define AT91C_SYSC_KEY        ((unsigned int) 0xFF << 24) // (RSTC) Password
450 // -------- SYSC_RSR : (RSTC Offset: 0x4) Reset Status Register --------
451 #  define AT91C_SYSC_URSTS      ((unsigned int) 0x1 <<  0)  // (RSTC) User
    … Reset Status
452 #  define AT91C_SYSC_BODSTS     ((unsigned int) 0x1 <<  1)  // (RSTC) Brown-out
    … Detection Status
453 #  define AT91C_SYSC_RSTTYP     ((unsigned int) 0x7 <<  8)  // (RSTC) Reset
    … Type
454 #  define   AT91C_SYSC_RSTTYP_POWERUP             ((unsigned int) 0x0 <<  8)
    … // (RSTC) Power-up Reset. VDDCORE rising.
455 #  define   AT91C_SYSC_RSTTYP_WATCHDOG            ((unsigned int) 0x2 <<  8)
    … // (RSTC) Watchdog Reset. Watchdog overflow occured.
456 #  define   AT91C_SYSC_RSTTYP_SOFTWARE            ((unsigned int) 0x3 <<  8)
    … // (RSTC) Software Reset. Processor reset required by the software.
457 #  define   AT91C_SYSC_RSTTYP_USER                ((unsigned int) 0x4 <<  8)
    … // (RSTC) User Reset. NRST pin detected low.
458 #  define   AT91C_SYSC_RSTTYP_BROWNOUT            ((unsigned int) 0x5 <<  8)
    … // (RSTC) Brown-out Reset.
459 #  define AT91C_SYSC_NRSTL      ((unsigned int) 0x1 << 16)  // (RSTC) NRST pin
    … level
460 #  define AT91C_SYSC_SRCMP      ((unsigned int) 0x1 << 17)  // (RSTC) Software
    … Reset Command in Progress.
461 // -------- SYSC_RMR : (RSTC Offset: 0x8) Reset Mode Register --------
462 #  define AT91C_SYSC_URSTEN     ((unsigned int) 0x1 <<  0)  // (RSTC) User
    … Reset Enable
463 #  define AT91C_SYSC_URSTIEN    ((unsigned int) 0x1 <<  4)  // (RSTC) User
    … Reset Interrupt Enable
```

```
464 #  define AT91C_SYSC_ERSTL      ((unsigned int) 0xF <<  8)  // (RSTC) User
    … Reset Enable
465 #  define AT91C_SYSC_BODIEN     ((unsigned int) 0x1 << 16)  // (RSTC) Brown-out
    … Detection Interrupt Enable
466
467 //
    ********************************************************************************
468 //               SOFTWARE API DEFINITION  FOR Real Time Timer Controller
    … Interface
469 //
    ********************************************************************************
470 typedef struct _AT91S_RTTC {
471   AT91_REG RTTC_RTMR;       // Real-time Mode Register
472   AT91_REG RTTC_RTAR;       // Real-time Alarm Register
473   AT91_REG RTTC_RTVR;       // Real-time Value Register
474   AT91_REG RTTC_RTSR;       // Real-time Status Register
475 } AT91S_RTTC, *AT91PS_RTTC;
476
477 // -------- SYSC_RTMR : (RTTC Offset: 0x0) Real-time Mode Register --------
478 #  define AT91C_SYSC_RTPRES     ((unsigned int) 0xFFFF <<  0)   // (RTTC)
    … Real-time Timer Prescaler Value
479 #  define AT91C_SYSC_ALMIEN     ((unsigned int) 0x1 << 16)  // (RTTC) Alarm
    … Interrupt Enable
480 #  define AT91C_SYSC_RTTINCIEN  ((unsigned int) 0x1 << 17)  // (RTTC) Real Time
    … Timer Increment Interrupt Enable
481 #  define AT91C_SYSC_RTTRST     ((unsigned int) 0x1 << 18)  // (RTTC) Real Time
    … Timer Restart
482 // -------- SYSC_RTAR : (RTTC Offset: 0x4) Real-time Alarm Register --------
483 #  define AT91C_SYSC_ALMV       ((unsigned int) 0x0 <<  0)  // (RTTC) Alarm
    … Value
484 // -------- SYSC_RTVR : (RTTC Offset: 0x8) Current Real-time Value Register
    … --------
485 #  define AT91C_SYSC_CRTV       ((unsigned int) 0x0 <<  0)  // (RTTC) Current
    … Real-time Value
486 // -------- SYSC_RTSR : (RTTC Offset: 0xc) Real-time Status Register --------
487 #  define AT91C_SYSC_ALMS       ((unsigned int) 0x1 <<  0)  // (RTTC) Real-time
    … Alarm Status
488 #  define AT91C_SYSC_RTTINC     ((unsigned int) 0x1 <<  1)  // (RTTC) Real-time
    … Timer Increment
489
```

```
490 //
    ********************************************************************************
491 //               SOFTWARE API DEFINITION  FOR Periodic Interval Timer Controller
    Interface
492 //
    ********************************************************************************
493 typedef struct _AT91S_PITC {
494   AT91_REG PITC_PIMR;       // Period Interval Mode Register
495   AT91_REG PITC_PISR;       // Period Interval Status Register
496   AT91_REG PITC_PIVR;       // Period Interval Value Register
497   AT91_REG PITC_PIIR;       // Period Interval Image Register
498 } AT91S_PITC, *AT91PS_PITC;
499
500 // -------- SYSC_PIMR : (PITC Offset: 0x0) Periodic Interval Mode Register
    --------
501 #  define AT91C_SYSC_PIV        ((unsigned int) 0xFFFFF <<  0)  // (PITC)
    Periodic Interval Value
502 #  define AT91C_SYSC_PITEN      ((unsigned int) 0x1 << 24)  // (PITC) Periodic
    Interval Timer Enabled
503 #  define AT91C_SYSC_PITIEN     ((unsigned int) 0x1 << 25)  // (PITC) Periodic
    Interval Timer Interrupt Enable
504 // -------- SYSC_PISR : (PITC Offset: 0x4) Periodic Interval Status Register
    --------
505 #  define AT91C_SYSC_PITS       ((unsigned int) 0x1 <<  0)  // (PITC) Periodic
    Interval Timer Status
506 // -------- SYSC_PIVR : (PITC Offset: 0x8) Periodic Interval Value Register
    --------
507 #  define AT91C_SYSC_CPIV       ((unsigned int) 0xFFFFF <<  0)  // (PITC)
    Current Periodic Interval Value
508 #  define AT91C_SYSC_PICNT      ((unsigned int) 0xFFF << 20)    // (PITC)
    Periodic Interval Counter
509 // -------- SYSC_PIIR : (PITC Offset: 0xc) Periodic Interval Image Register
    --------
510
511 //
    ********************************************************************************
512 //               SOFTWARE API DEFINITION  FOR Watchdog Timer Controller
    Interface
513 //
    ********************************************************************************
```

```
514 typedef struct _AT91S_WDTC {
515   AT91_REG WDTC_WDCR;       // Watchdog Control Register
516   AT91_REG WDTC_WDMR;       // Watchdog Mode Register
517   AT91_REG WDTC_WDSR;       // Watchdog Status Register
518 } AT91S_WDTC, *AT91PS_WDTC;
519
520 // -------- SYSC_WDCR : (WDTC Offset: 0x0) Periodic Interval Image Register
    --------
521 #  define AT91C_SYSC_WDRSTT     ((unsigned int) 0x1 <<  0)  // (WDTC) Watchdog
    Restart
522 // -------- SYSC_WDMR : (WDTC Offset: 0x4) Watchdog Mode Register --------
523 #  define AT91C_SYSC_WDV        ((unsigned int) 0xFFF <<  0)    // (WDTC)
    Watchdog Timer Restart
524 #  define AT91C_SYSC_WDFIEN     ((unsigned int) 0x1 << 12)  // (WDTC) Watchdog
    Fault Interrupt Enable
525 #  define AT91C_SYSC_WDRSTEN    ((unsigned int) 0x1 << 13)  // (WDTC) Watchdog
    Reset Enable
526 #  define AT91C_SYSC_WDRPROC    ((unsigned int) 0x1 << 14)  // (WDTC) Watchdog
    Timer Restart
527 #  define AT91C_SYSC_WDDIS      ((unsigned int) 0x1 << 15)  // (WDTC) Watchdog
    Disable
528 #  define AT91C_SYSC_WDD        ((unsigned int) 0xFFF << 16)    // (WDTC)
    Watchdog Delta Value
529 #  define AT91C_SYSC_WDDBGHLT   ((unsigned int) 0x1 << 28)  // (WDTC) Watchdog
    Debug Halt
530 #  define AT91C_SYSC_WDIDLEHLT  ((unsigned int) 0x1 << 29)  // (WDTC) Watchdog
    Idle Halt
531 // -------- SYSC_WDSR : (WDTC Offset: 0x8) Watchdog Status Register --------
532 #  define AT91C_SYSC_WDUNF      ((unsigned int) 0x1 <<  0)  // (WDTC) Watchdog
    Underflow
533 #  define AT91C_SYSC_WDERR      ((unsigned int) 0x1 <<  1)  // (WDTC) Watchdog
    Error
534
535 //
    ******************************************************************************
536 //               SOFTWARE API DEFINITION  FOR Memory Controller Interface
537 //
    ******************************************************************************
538 typedef struct _AT91S_MC {
539   AT91_REG MC_RCR;      // MC Remap Control Register
```

```
540   AT91_REG MC_ASR;       // MC Abort Status Register
541   AT91_REG MC_AASR;      // MC Abort Address Status Register
542   AT91_REG Reserved0[21];   //
543   AT91_REG MC_FMR;       // MC Flash Mode Register
544   AT91_REG MC_FCR;       // MC Flash Command Register
545   AT91_REG MC_FSR;       // MC Flash Status Register
546 } AT91S_MC, *AT91PS_MC;
547
548 // -------- MC_RCR : (MC Offset: 0x0) MC Remap Control Register --------
549 #  define AT91C_MC_RCB          ((unsigned int) 0x1 <<  0)  // (MC) Remap
    … Command Bit
550 // -------- MC_ASR : (MC Offset: 0x4) MC Abort Status Register --------
551 #  define AT91C_MC_UNDADD       ((unsigned int) 0x1 <<  0)  // (MC) Undefined
    … Addess Abort Status
552 #  define AT91C_MC_MISADD       ((unsigned int) 0x1 <<  1)  // (MC) Misaligned
    … Addess Abort Status
553 #  define AT91C_MC_ABTSZ        ((unsigned int) 0x3 <<  8)  // (MC) Abort Size
    … Status
554 #  define   AT91C_MC_ABTSZ_BYTE                 ((unsigned int) 0x0 <<  8)  //
    … (MC) Byte
555 #  define   AT91C_MC_ABTSZ_HWORD                ((unsigned int) 0x1 <<  8)  //
    … (MC) Half-word
556 #  define   AT91C_MC_ABTSZ_WORD                 ((unsigned int) 0x2 <<  8)  //
    … (MC) Word
557 #  define AT91C_MC_ABTTYP       ((unsigned int) 0x3 << 10)  // (MC) Abort Type
    … Status
558 #  define   AT91C_MC_ABTTYP_DATAR                ((unsigned int) 0x0 << 10) //
    … (MC) Data Read
559 #  define   AT91C_MC_ABTTYP_DATAW                ((unsigned int) 0x1 << 10) //
    … (MC) Data Write
560 #  define   AT91C_MC_ABTTYP_FETCH                ((unsigned int) 0x2 << 10) //
    … (MC) Code Fetch
561 #  define AT91C_MC_MST0         ((unsigned int) 0x1 << 16)  // (MC) Master 0
    … Abort Source
562 #  define AT91C_MC_MST1         ((unsigned int) 0x1 << 17)  // (MC) Master 1
    … Abort Source
563 #  define AT91C_MC_SVMST0       ((unsigned int) 0x1 << 24)  // (MC) Saved
    … Master 0 Abort Source
564 #  define AT91C_MC_SVMST1       ((unsigned int) 0x1 << 25)  // (MC) Saved
    … Master 1 Abort Source
```

```
565 // -------- MC_FMR : (MC Offset: 0x60) MC Flash Mode Register --------
566 #  define AT91C_MC_FRDY         ((unsigned int) 0x1 <<  0)  // (MC) Flash Ready
567 #  define AT91C_MC_LOCKE        ((unsigned int) 0x1 <<  2)  // (MC) Lock Error
568 #  define AT91C_MC_PROGE        ((unsigned int) 0x1 <<  3)  // (MC) Programming
    … Error
569 #  define AT91C_MC_NEBP         ((unsigned int) 0x1 <<  7)  // (MC) No Erase
    … Before Programming
570 #  define AT91C_MC_FWS          ((unsigned int) 0x3 <<  8)  // (MC) Flash Wait
    … State
571 #  define   AT91C_MC_FWS_0FWS                 ((unsigned int) 0x0 <<  8)    //
    … (MC) 1 cycle for Read, 2 for Write operations
572 #  define   AT91C_MC_FWS_1FWS                 ((unsigned int) 0x1 <<  8)    //
    … (MC) 2 cycles for Read, 3 for Write operations
573 #  define   AT91C_MC_FWS_2FWS                 ((unsigned int) 0x2 <<  8)    //
    … (MC) 3 cycles for Read, 4 for Write operations
574 #  define   AT91C_MC_FWS_3FWS                 ((unsigned int) 0x3 <<  8)    //
    … (MC) 4 cycles for Read, 4 for Write operations
575 #  define AT91C_MC_FMCN         ((unsigned int) 0xFF << 16) // (MC) Flash
    … Microsecond Cycle Number
576 // -------- MC_FCR : (MC Offset: 0x64) MC Flash Command Register --------
577 #  define AT91C_MC_FCMD         ((unsigned int) 0xF <<  0)  // (MC) Flash
    … Command
578 #  define   AT91C_MC_FCMD_START_PROG          ((unsigned int) 0x1) // (MC)
    … Starts the programming of th epage specified by PAGEN.
579 #  define   AT91C_MC_FCMD_LOCK                ((unsigned int) 0x2) // (MC)
    … Starts a lock sequence of the sector defined by the bits 4 to 7 of the field
    … PAGEN.
580 #  define   AT91C_MC_FCMD_PROG_AND_LOCK       ((unsigned int) 0x3) // (MC) The
    … lock sequence automatically happens after the programming sequence is
    … completed.
581 #  define   AT91C_MC_FCMD_UNLOCK              ((unsigned int) 0x4) // (MC)
    … Starts an unlock sequence of the sector defined by the bits 4 to 7 of the field
    … PAGEN.
582 #  define   AT91C_MC_FCMD_ERASE_ALL           ((unsigned int) 0x8) // (MC)
    … Starts the erase of the entire flash.If at least a page is locked, the command
    … is cancelled.
583 #  define   AT91C_MC_FCMD_SET_GP_NVM          ((unsigned int) 0xB) // (MC) Set
    … General Purpose NVM bits.
584 #  define   AT91C_MC_FCMD_CLR_GP_NVM          ((unsigned int) 0xD) // (MC)
    … Clear General Purpose NVM bits.
```

```
585 #  define   AT91C_MC_FCMD_SET_SECURITY        ((unsigned int) 0xF) // (MC) Set
    … Security Bit.
586 #  define AT91C_MC_PAGEN        ((unsigned int) 0x3FF <<  8)    // (MC) Page
    … Number
587 #  define AT91C_MC_KEY          ((unsigned int) 0xFF << 24) // (MC) Writing
    … Protect Key
588 // -------- MC_FSR : (MC Offset: 0x68) MC Flash Command Register --------
589 #  define AT91C_MC_SECURITY     ((unsigned int) 0x1 <<  4)  // (MC) Security
    … Bit Status
590 #  define AT91C_MC_GPNVM0       ((unsigned int) 0x1 <<  8)  // (MC) Sector 0
    … Lock Status
591 #  define AT91C_MC_GPNVM1       ((unsigned int) 0x1 <<  9)  // (MC) Sector 1
    … Lock Status
592 #  define AT91C_MC_GPNVM2       ((unsigned int) 0x1 << 10)  // (MC) Sector 2
    … Lock Status
593 #  define AT91C_MC_GPNVM3       ((unsigned int) 0x1 << 11)  // (MC) Sector 3
    … Lock Status
594 #  define AT91C_MC_GPNVM4       ((unsigned int) 0x1 << 12)  // (MC) Sector 4
    … Lock Status
595 #  define AT91C_MC_GPNVM5       ((unsigned int) 0x1 << 13)  // (MC) Sector 5
    … Lock Status
596 #  define AT91C_MC_GPNVM6       ((unsigned int) 0x1 << 14)  // (MC) Sector 6
    … Lock Status
597 #  define AT91C_MC_GPNVM7       ((unsigned int) 0x1 << 15)  // (MC) Sector 7
    … Lock Status
598 #  define AT91C_MC_LOCKS0       ((unsigned int) 0x1 << 16)  // (MC) Sector 0
    … Lock Status
599 #  define AT91C_MC_LOCKS1       ((unsigned int) 0x1 << 17)  // (MC) Sector 1
    … Lock Status
600 #  define AT91C_MC_LOCKS2       ((unsigned int) 0x1 << 18)  // (MC) Sector 2
    … Lock Status
601 #  define AT91C_MC_LOCKS3       ((unsigned int) 0x1 << 19)  // (MC) Sector 3
    … Lock Status
602 #  define AT91C_MC_LOCKS4       ((unsigned int) 0x1 << 20)  // (MC) Sector 4
    … Lock Status
603 #  define AT91C_MC_LOCKS5       ((unsigned int) 0x1 << 21)  // (MC) Sector 5
    … Lock Status
604 #  define AT91C_MC_LOCKS6       ((unsigned int) 0x1 << 22)  // (MC) Sector 6
    … Lock Status
605 #  define AT91C_MC_LOCKS7       ((unsigned int) 0x1 << 23)  // (MC) Sector 7
```

```
605… Lock Status
606 #  define AT91C_MC_LOCKS8       ((unsigned int) 0x1 << 24)  // (MC) Sector 8
  … Lock Status
607 #  define AT91C_MC_LOCKS9       ((unsigned int) 0x1 << 25)  // (MC) Sector 9
  … Lock Status
608 #  define AT91C_MC_LOCKS10      ((unsigned int) 0x1 << 26)  // (MC) Sector 10
  … Lock Status
609 #  define AT91C_MC_LOCKS11      ((unsigned int) 0x1 << 27)  // (MC) Sector 11
  … Lock Status
610 #  define AT91C_MC_LOCKS12      ((unsigned int) 0x1 << 28)  // (MC) Sector 12
  … Lock Status
611 #  define AT91C_MC_LOCKS13      ((unsigned int) 0x1 << 29)  // (MC) Sector 13
  … Lock Status
612 #  define AT91C_MC_LOCKS14      ((unsigned int) 0x1 << 30)  // (MC) Sector 14
  … Lock Status
613 #  define AT91C_MC_LOCKS15      ((unsigned int) 0x1 << 31)  // (MC) Sector 15
  … Lock Status
614
615 //
  … ********************************************************************************
616 //               SOFTWARE API DEFINITION  FOR Serial Parallel Interface
617 //
  … ********************************************************************************
618 typedef struct _AT91S_SPI {
619   AT91_REG SPI_CR;       // Control Register
620   AT91_REG SPI_MR;       // Mode Register
621   AT91_REG SPI_RDR;      // Receive Data Register
622   AT91_REG SPI_TDR;      // Transmit Data Register
623   AT91_REG SPI_SR;       // Status Register
624   AT91_REG SPI_IER;      // Interrupt Enable Register
625   AT91_REG SPI_IDR;      // Interrupt Disable Register
626   AT91_REG SPI_IMR;      // Interrupt Mask Register
627   AT91_REG Reserved0[4];    //
628   AT91_REG SPI_CSR[4];      // Chip Select Register
629   AT91_REG Reserved1[48];   //
630   AT91_REG SPI_RPR;      // Receive Pointer Register
631   AT91_REG SPI_RCR;      // Receive Counter Register
632   AT91_REG SPI_TPR;      // Transmit Pointer Register
633   AT91_REG SPI_TCR;      // Transmit Counter Register
634   AT91_REG SPI_RNPR;        // Receive Next Pointer Register
```

```
635   AT91_REG SPI_RNCR;        // Receive Next Counter Register
636   AT91_REG SPI_TNPR;        // Transmit Next Pointer Register
637   AT91_REG SPI_TNCR;        // Transmit Next Counter Register
638   AT91_REG SPI_PTCR;        // PDC Transfer Control Register
639   AT91_REG SPI_PTSR;        // PDC Transfer Status Register
640 } AT91S_SPI, *AT91PS_SPI;
641 
642 // -------- SPI_CR : (SPI Offset: 0x0) SPI Control Register --------
643 #  define AT91C_SPI_SPIEN       ((unsigned int) 0x1 <<  0)  // (SPI) SPI Enable
644 #  define AT91C_SPI_SPIDIS      ((unsigned int) 0x1 <<  1)  // (SPI) SPI
    … Disable
645 #  define AT91C_SPI_SWRST       ((unsigned int) 0x1 <<  7)  // (SPI) SPI
    … Software reset
646 #  define AT91C_SPI_LASTXFER    ((unsigned int) 0x1 << 24)  // (SPI) SPI Last
    … Transfer
647 // -------- SPI_MR : (SPI Offset: 0x4) SPI Mode Register --------
648 #  define AT91C_SPI_MSTR        ((unsigned int) 0x1 <<  0)  // (SPI)
    … Master/Slave Mode
649 #  define AT91C_SPI_PS          ((unsigned int) 0x1 <<  1)  // (SPI) Peripheral
    … Select
650 #  define   AT91C_SPI_PS_FIXED                ((unsigned int) 0x0 <<  1)    //
    … (SPI) Fixed Peripheral Select
651 #  define   AT91C_SPI_PS_VARIABLE             ((unsigned int) 0x1 <<  1)    //
    … (SPI) Variable Peripheral Select
652 #  define AT91C_SPI_PCSDEC      ((unsigned int) 0x1 <<  2)  // (SPI) Chip
    … Select Decode
653 #  define AT91C_SPI_FDIV        ((unsigned int) 0x1 <<  3)  // (SPI) Clock
    … Selection
654 #  define AT91C_SPI_MODFDIS     ((unsigned int) 0x1 <<  4)  // (SPI) Mode Fault
    … Detection
655 #  define AT91C_SPI_LLB         ((unsigned int) 0x1 <<  7)  // (SPI) Clock
    … Selection
656 #  define AT91C_SPI_PCS         ((unsigned int) 0xF << 16)  // (SPI) Peripheral
    … Chip Select
657 #  define AT91C_SPI_DLYBCS      ((unsigned int) 0xFF << 24) // (SPI) Delay
    … Between Chip Selects
658 // -------- SPI_RDR : (SPI Offset: 0x8) Receive Data Register --------
659 #  define AT91C_SPI_RD          ((unsigned int) 0xFFFF <<  0)   // (SPI)
    … Receive Data
660 #  define AT91C_SPI_RPCS        ((unsigned int) 0xF << 16)  // (SPI) Peripheral
```

```
660… Chip Select Status
661 // -------- SPI_TDR : (SPI Offset: 0xc) Transmit Data Register --------
662 #  define AT91C_SPI_TD          ((unsigned int) 0xFFFF <<  0)   // (SPI)
… Transmit Data
663 #  define AT91C_SPI_TPCS        ((unsigned int) 0xF << 16)  // (SPI) Peripheral
… Chip Select Status
664 // -------- SPI_SR : (SPI Offset: 0x10) Status Register --------
665 #  define AT91C_SPI_RDRF        ((unsigned int) 0x1 <<  0)  // (SPI) Receive
… Data Register Full
666 #  define AT91C_SPI_TDRE        ((unsigned int) 0x1 <<  1)  // (SPI) Transmit
… Data Register Empty
667 #  define AT91C_SPI_MODF        ((unsigned int) 0x1 <<  2)  // (SPI) Mode Fault
… Error
668 #  define AT91C_SPI_OVRES       ((unsigned int) 0x1 <<  3)  // (SPI) Overrun
… Error Status
669 #  define AT91C_SPI_ENDRX       ((unsigned int) 0x1 <<  4)  // (SPI) End of
… Receiver Transfer
670 #  define AT91C_SPI_ENDTX       ((unsigned int) 0x1 <<  5)  // (SPI) End of
… Receiver Transfer
671 #  define AT91C_SPI_RXBUFF      ((unsigned int) 0x1 <<  6)  // (SPI) RXBUFF
… Interrupt
672 #  define AT91C_SPI_TXBUFE      ((unsigned int) 0x1 <<  7)  // (SPI) TXBUFE
… Interrupt
673 #  define AT91C_SPI_NSSR        ((unsigned int) 0x1 <<  8)  // (SPI) NSSR
… Interrupt
674 #  define AT91C_SPI_TXEMPTY     ((unsigned int) 0x1 <<  9)  // (SPI) TXEMPTY
… Interrupt
675 #  define AT91C_SPI_SPIENS      ((unsigned int) 0x1 << 16)  // (SPI) Enable
… Status
676 // -------- SPI_IER : (SPI Offset: 0x14) Interrupt Enable Register --------
677 // -------- SPI_IDR : (SPI Offset: 0x18) Interrupt Disable Register --------
678 // -------- SPI_IMR : (SPI Offset: 0x1c) Interrupt Mask Register --------
679 // -------- SPI_CSR : (SPI Offset: 0x30) Chip Select Register --------
680 #  define AT91C_SPI_CPOL        ((unsigned int) 0x1 <<  0)  // (SPI) Clock
… Polarity
681 #  define AT91C_SPI_NCPHA       ((unsigned int) 0x1 <<  1)  // (SPI) Clock
… Phase
682 #  define AT91C_SPI_CSAAT       ((unsigned int) 0x1 <<  2)  // (SPI) Chip
… Select Active After Transfer
683 #  define AT91C_SPI_BITS        ((unsigned int) 0xF <<  4)  // (SPI) Bits Per
```

```
683… Transfer
684 #  define   AT91C_SPI_BITS_8                    ((unsigned int) 0x0 <<  4)  //
… (SPI) 8 Bits Per transfer
685 #  define   AT91C_SPI_BITS_9                    ((unsigned int) 0x1 <<  4)  //
… (SPI) 9 Bits Per transfer
686 #  define   AT91C_SPI_BITS_10                   ((unsigned int) 0x2 <<  4)  //
… (SPI) 10 Bits Per transfer
687 #  define   AT91C_SPI_BITS_11                   ((unsigned int) 0x3 <<  4)  //
… (SPI) 11 Bits Per transfer
688 #  define   AT91C_SPI_BITS_12                   ((unsigned int) 0x4 <<  4)  //
… (SPI) 12 Bits Per transfer
689 #  define   AT91C_SPI_BITS_13                   ((unsigned int) 0x5 <<  4)  //
… (SPI) 13 Bits Per transfer
690 #  define   AT91C_SPI_BITS_14                   ((unsigned int) 0x6 <<  4)  //
… (SPI) 14 Bits Per transfer
691 #  define   AT91C_SPI_BITS_15                   ((unsigned int) 0x7 <<  4)  //
… (SPI) 15 Bits Per transfer
692 #  define   AT91C_SPI_BITS_16                   ((unsigned int) 0x8 <<  4)  //
… (SPI) 16 Bits Per transfer
693 #  define AT91C_SPI_SCBR        ((unsigned int) 0xFF <<  8) // (SPI) Serial
… Clock Baud Rate
694 #  define AT91C_SPI_DLYBS       ((unsigned int) 0xFF << 16) // (SPI) Serial
… Clock Baud Rate
695 #  define AT91C_SPI_DLYBCT      ((unsigned int) 0xFF << 24) // (SPI) Delay
… Between Consecutive Transfers
696
697 //
… ////////////////////////////////////////////////////////////////////////////
698 //              SOFTWARE API DEFINITION  FOR Analog to Digital Convertor
699 //
… ////////////////////////////////////////////////////////////////////////////
700 typedef struct _AT91S_ADC {
701   AT91_REG ADC_CR;      // ADC Control Register
702   AT91_REG ADC_MR;      // ADC Mode Register
703   AT91_REG Reserved0[2];    //
704   AT91_REG ADC_CHER;        // ADC Channel Enable Register
705   AT91_REG ADC_CHDR;        // ADC Channel Disable Register
706   AT91_REG ADC_CHSR;        // ADC Channel Status Register
707   AT91_REG ADC_SR;      // ADC Status Register
708   AT91_REG ADC_LCDR;        // ADC Last Converted Data Register
```

```
709   AT91_REG ADC_IER;     // ADC Interrupt Enable Register
710   AT91_REG ADC_IDR;     // ADC Interrupt Disable Register
711   AT91_REG ADC_IMR;     // ADC Interrupt Mask Register
712   AT91_REG ADC_CDR0;        // ADC Channel Data Register 0
713   AT91_REG ADC_CDR1;        // ADC Channel Data Register 1
714   AT91_REG ADC_CDR2;        // ADC Channel Data Register 2
715   AT91_REG ADC_CDR3;        // ADC Channel Data Register 3
716   AT91_REG ADC_CDR4;        // ADC Channel Data Register 4
717   AT91_REG ADC_CDR5;        // ADC Channel Data Register 5
718   AT91_REG ADC_CDR6;        // ADC Channel Data Register 6
719   AT91_REG ADC_CDR7;        // ADC Channel Data Register 7
720   AT91_REG Reserved1[44];   //
721   AT91_REG ADC_RPR;     // Receive Pointer Register
722   AT91_REG ADC_RCR;     // Receive Counter Register
723   AT91_REG ADC_TPR;     // Transmit Pointer Register
724   AT91_REG ADC_TCR;     // Transmit Counter Register
725   AT91_REG ADC_RNPR;        // Receive Next Pointer Register
726   AT91_REG ADC_RNCR;        // Receive Next Counter Register
727   AT91_REG ADC_TNPR;        // Transmit Next Pointer Register
728   AT91_REG ADC_TNCR;        // Transmit Next Counter Register
729   AT91_REG ADC_PTCR;        // PDC Transfer Control Register
730   AT91_REG ADC_PTSR;        // PDC Transfer Status Register
731 } AT91S_ADC, *AT91PS_ADC;
732
733 // -------- ADC_CR : (ADC Offset: 0x0) ADC Control Register --------
734 #  define AT91C_ADC_SWRST       ((unsigned int) 0x1 <<  0)  // (ADC) Software
    … Reset
735 #  define AT91C_ADC_START       ((unsigned int) 0x1 <<  1)  // (ADC) Start
    … Conversion
736 // -------- ADC_MR : (ADC Offset: 0x4) ADC Mode Register --------
737 #  define AT91C_ADC_TRGEN       ((unsigned int) 0x1 <<  0)  // (ADC) Trigger
    … Enable
738 #  define   AT91C_ADC_TRGEN_DIS                 ((unsigned int) 0x0)   //
    … (ADC) Hradware triggers are disabled. Starting a conversion is only possible by
    … software
739 #  define   AT91C_ADC_TRGEN_EN                  ((unsigned int) 0x1)   //
    … (ADC) Hardware trigger selected by TRGSEL field is enabled.
740 #  define AT91C_ADC_TRGSEL      ((unsigned int) 0x7 <<  1)  // (ADC) Trigger
    … Selection
741 #  define   AT91C_ADC_TRGSEL_TIOA0              ((unsigned int) 0x0 <<  1)
```

```
741… // (ADC) Selected TRGSEL = TIAO0
742  #  define   AT91C_ADC_TRGSEL_TIOA1              ((unsigned int) 0x1 <<  1)
…    // (ADC) Selected TRGSEL = TIAO1
743  #  define   AT91C_ADC_TRGSEL_TIOA2              ((unsigned int) 0x2 <<  1)
…    // (ADC) Selected TRGSEL = TIAO2
744  #  define   AT91C_ADC_TRGSEL_TIOA3              ((unsigned int) 0x3 <<  1)
…    // (ADC) Selected TRGSEL = TIAO3
745  #  define   AT91C_ADC_TRGSEL_TIOA4              ((unsigned int) 0x4 <<  1)
…    // (ADC) Selected TRGSEL = TIAO4
746  #  define   AT91C_ADC_TRGSEL_TIOA5              ((unsigned int) 0x5 <<  1)
…    // (ADC) Selected TRGSEL = TIAO5
747  #  define   AT91C_ADC_TRGSEL_EXT                ((unsigned int) 0x6 <<  1)
…    // (ADC) Selected TRGSEL = External Trigger
748  #  define AT91C_ADC_LOWRES      ((unsigned int) 0x1 <<  4)  // (ADC)
…    Resolution.
749  #  define   AT91C_ADC_LOWRES_10_BIT             ((unsigned int) 0x0 <<  4)
…    // (ADC) 10-bit resolution
750  #  define   AT91C_ADC_LOWRES_8_BIT              ((unsigned int) 0x1 <<  4)
…    // (ADC) 8-bit resolution
751  #  define AT91C_ADC_SLEEP       ((unsigned int) 0x1 <<  5)  // (ADC) Sleep Mode
752  #  define   AT91C_ADC_SLEEP_NORMAL_MODE          ((unsigned int) 0x0 <<  5) //
…    (ADC) Normal Mode
753  #  define   AT91C_ADC_SLEEP_MODE                ((unsigned int) 0x1 <<  5) //
…    (ADC) Sleep Mode
754  #  define AT91C_ADC_PRESCAL     ((unsigned int) 0x3F <<  8) // (ADC) Prescaler
…    rate selection
755  #  define AT91C_ADC_STARTUP     ((unsigned int) 0x1F << 16) // (ADC) Startup
…    Time
756  #  define AT91C_ADC_SHTIM       ((unsigned int) 0xF << 24)  // (ADC) Sample &
…    Hold Time
757  // --------     ADC_CHER : (ADC Offset: 0x10) ADC Channel Enable Register
…    --------
758  #  define AT91C_ADC_CH0         ((unsigned int) 0x1 <<  0)  // (ADC) Channel 0
759  #  define AT91C_ADC_CH1         ((unsigned int) 0x1 <<  1)  // (ADC) Channel 1
760  #  define AT91C_ADC_CH2         ((unsigned int) 0x1 <<  2)  // (ADC) Channel 2
761  #  define AT91C_ADC_CH3         ((unsigned int) 0x1 <<  3)  // (ADC) Channel 3
762  #  define AT91C_ADC_CH4         ((unsigned int) 0x1 <<  4)  // (ADC) Channel 4
763  #  define AT91C_ADC_CH5         ((unsigned int) 0x1 <<  5)  // (ADC) Channel 5
764  #  define AT91C_ADC_CH6         ((unsigned int) 0x1 <<  6)  // (ADC) Channel 6
765  #  define AT91C_ADC_CH7         ((unsigned int) 0x1 <<  7)  // (ADC) Channel 7
```

```
766 // --------     ADC_CHDR : (ADC Offset: 0x14) ADC Channel Disable Register
    --------
767 // --------     ADC_CHSR : (ADC Offset: 0x18) ADC Channel Status Register
    --------
768 // -------- ADC_SR : (ADC Offset: 0x1c) ADC Status Register --------
769 #  define AT91C_ADC_EOC0        ((unsigned int) 0x1 <<  0)  // (ADC) End of
    Conversion
770 #  define AT91C_ADC_EOC1        ((unsigned int) 0x1 <<  1)  // (ADC) End of
    Conversion
771 #  define AT91C_ADC_EOC2        ((unsigned int) 0x1 <<  2)  // (ADC) End of
    Conversion
772 #  define AT91C_ADC_EOC3        ((unsigned int) 0x1 <<  3)  // (ADC) End of
    Conversion
773 #  define AT91C_ADC_EOC4        ((unsigned int) 0x1 <<  4)  // (ADC) End of
    Conversion
774 #  define AT91C_ADC_EOC5        ((unsigned int) 0x1 <<  5)  // (ADC) End of
    Conversion
775 #  define AT91C_ADC_EOC6        ((unsigned int) 0x1 <<  6)  // (ADC) End of
    Conversion
776 #  define AT91C_ADC_EOC7        ((unsigned int) 0x1 <<  7)  // (ADC) End of
    Conversion
777 #  define AT91C_ADC_OVRE0       ((unsigned int) 0x1 <<  8)  // (ADC) Overrun
    Error
778 #  define AT91C_ADC_OVRE1       ((unsigned int) 0x1 <<  9)  // (ADC) Overrun
    Error
779 #  define AT91C_ADC_OVRE2       ((unsigned int) 0x1 << 10)  // (ADC) Overrun
    Error
780 #  define AT91C_ADC_OVRE3       ((unsigned int) 0x1 << 11)  // (ADC) Overrun
    Error
781 #  define AT91C_ADC_OVRE4       ((unsigned int) 0x1 << 12)  // (ADC) Overrun
    Error
782 #  define AT91C_ADC_OVRE5       ((unsigned int) 0x1 << 13)  // (ADC) Overrun
    Error
783 #  define AT91C_ADC_OVRE6       ((unsigned int) 0x1 << 14)  // (ADC) Overrun
    Error
784 #  define AT91C_ADC_OVRE7       ((unsigned int) 0x1 << 15)  // (ADC) Overrun
    Error
785 #  define AT91C_ADC_DRDY        ((unsigned int) 0x1 << 16)  // (ADC) Data Ready
786 #  define AT91C_ADC_GOVRE       ((unsigned int) 0x1 << 17)  // (ADC) General
    Overrun
```

```
#  define AT91C_ADC_ENDRX       ((unsigned int) 0x1 << 18)  // (ADC) End of Receiver Transfer
#  define AT91C_ADC_RXBUFF      ((unsigned int) 0x1 << 19)  // (ADC) RXBUFF Interrupt
// -------- ADC_LCDR : (ADC Offset: 0x20) ADC Last Converted Data Register --------
#  define AT91C_ADC_LDATA       ((unsigned int) 0x3FF <<  0)    // (ADC) Last Data Converted
// -------- ADC_IER : (ADC Offset: 0x24) ADC Interrupt Enable Register --------
// -------- ADC_IDR : (ADC Offset: 0x28) ADC Interrupt Disable Register --------
// -------- ADC_IMR : (ADC Offset: 0x2c) ADC Interrupt Mask Register --------
// -------- ADC_CDR0 : (ADC Offset: 0x30) ADC Channel Data Register 0 --------
#  define AT91C_ADC_DATA        ((unsigned int) 0x3FF <<  0)    // (ADC) Converted Data
// -------- ADC_CDR1 : (ADC Offset: 0x34) ADC Channel Data Register 1 --------
// -------- ADC_CDR2 : (ADC Offset: 0x38) ADC Channel Data Register 2 --------
// -------- ADC_CDR3 : (ADC Offset: 0x3c) ADC Channel Data Register 3 --------
// -------- ADC_CDR4 : (ADC Offset: 0x40) ADC Channel Data Register 4 --------
// -------- ADC_CDR5 : (ADC Offset: 0x44) ADC Channel Data Register 5 --------
// -------- ADC_CDR6 : (ADC Offset: 0x48) ADC Channel Data Register 6 --------
// -------- ADC_CDR7 : (ADC Offset: 0x4c) ADC Channel Data Register 7 --------

//
//  ////////////////////////////////////////////////////////////////////////////
//               SOFTWARE API DEFINITION  FOR Synchronous Serial Controller Interface
//
//  ////////////////////////////////////////////////////////////////////////////
typedef struct _AT91S_SSC {
  AT91_REG SSC_CR;      // Control Register
  AT91_REG SSC_CMR;     // Clock Mode Register
  AT91_REG Reserved0[2];    //
  AT91_REG SSC_RCMR;        // Receive Clock ModeRegister
  AT91_REG SSC_RFMR;        // Receive Frame Mode Register
  AT91_REG SSC_TCMR;        // Transmit Clock Mode Register
  AT91_REG SSC_TFMR;        // Transmit Frame Mode Register
  AT91_REG SSC_RHR;     // Receive Holding Register
  AT91_REG SSC_THR;     // Transmit Holding Register
  AT91_REG Reserved1[2];    //
```

```
818   AT91_REG SSC_RSHR;        // Receive Sync Holding Register
819   AT91_REG SSC_TSHR;        // Transmit Sync Holding Register
820   AT91_REG SSC_RC0R;        // Receive Compare 0 Register
821   AT91_REG SSC_RC1R;        // Receive Compare 1 Register
822   AT91_REG SSC_SR;      // Status Register
823   AT91_REG SSC_IER;     // Interrupt Enable Register
824   AT91_REG SSC_IDR;     // Interrupt Disable Register
825   AT91_REG SSC_IMR;     // Interrupt Mask Register
826   AT91_REG Reserved2[44];   //
827   AT91_REG SSC_RPR;     // Receive Pointer Register
828   AT91_REG SSC_RCR;     // Receive Counter Register
829   AT91_REG SSC_TPR;     // Transmit Pointer Register
830   AT91_REG SSC_TCR;     // Transmit Counter Register
831   AT91_REG SSC_RNPR;        // Receive Next Pointer Register
832   AT91_REG SSC_RNCR;        // Receive Next Counter Register
833   AT91_REG SSC_TNPR;        // Transmit Next Pointer Register
834   AT91_REG SSC_TNCR;        // Transmit Next Counter Register
835   AT91_REG SSC_PTCR;        // PDC Transfer Control Register
836   AT91_REG SSC_PTSR;        // PDC Transfer Status Register
837 } AT91S_SSC, *AT91PS_SSC;
838
839 // -------- SSC_CR : (SSC Offset: 0x0) SSC Control Register --------
840 #  define AT91C_SSC_RXEN        ((unsigned int) 0x1 <<  0)  // (SSC) Receive
    … Enable
841 #  define AT91C_SSC_RXDIS       ((unsigned int) 0x1 <<  1)  // (SSC) Receive
    … Disable
842 #  define AT91C_SSC_TXEN        ((unsigned int) 0x1 <<  8)  // (SSC) Transmit
    … Enable
843 #  define AT91C_SSC_TXDIS       ((unsigned int) 0x1 <<  9)  // (SSC) Transmit
    … Disable
844 #  define AT91C_SSC_SWRST       ((unsigned int) 0x1 << 15)  // (SSC) Software
    … Reset
845 // -------- SSC_RCMR : (SSC Offset: 0x10) SSC Receive Clock Mode Register
    … --------
846 #  define AT91C_SSC_CKS         ((unsigned int) 0x3 <<  0)  // (SSC)
    … Receive/Transmit Clock Selection
847 #  define   AT91C_SSC_CKS_DIV                 ((unsigned int) 0x0) // (SSC)
    … Divided Clock
848 #  define   AT91C_SSC_CKS_TK                  ((unsigned int) 0x1) // (SSC) TK
    … Clock signal
```

```
849 #  define   AT91C_SSC_CKS_RK                  ((unsigned int) 0x2) // (SSC) RK
… pin
850 #  define AT91C_SSC_CKO         ((unsigned int) 0x7 <<  2)  // (SSC)
… Receive/Transmit Clock Output Mode Selection
851 #  define   AT91C_SSC_CKO_NONE                ((unsigned int) 0x0 <<  2)   //
… (SSC) Receive/Transmit Clock Output Mode: None RK pin: Input-only
852 #  define   AT91C_SSC_CKO_CONTINOUS           ((unsigned int) 0x1 <<  2)   //
… (SSC) Continuous Receive/Transmit Clock RK pin: Output
853 #  define   AT91C_SSC_CKO_DATA_TX             ((unsigned int) 0x2 <<  2)   //
… (SSC) Receive/Transmit Clock only during data transfers RK pin: Output
854 #  define AT91C_SSC_CKI         ((unsigned int) 0x1 <<  5)  // (SSC)
… Receive/Transmit Clock Inversion
855 #  define AT91C_SSC_CKG         ((unsigned int) 0x3 <<  6)  // (SSC)
… Receive/Transmit Clock Gating Selection
856 #  define   AT91C_SSC_CKG_NONE                ((unsigned int) 0x0 <<  6)   //
… (SSC) Receive/Transmit Clock Gating: None, continuous clock
857 #  define   AT91C_SSC_CKG_LOW                 ((unsigned int) 0x1 <<  6)   //
… (SSC) Receive/Transmit Clock enabled only if RF Low
858 #  define   AT91C_SSC_CKG_HIGH                ((unsigned int) 0x2 <<  6)   //
… (SSC) Receive/Transmit Clock enabled only if RF High
859 #  define AT91C_SSC_START       ((unsigned int) 0xF <<  8)  // (SSC)
… Receive/Transmit Start Selection
860 #  define   AT91C_SSC_START_CONTINOUS           ((unsigned int) 0x0 <<  8) //
… (SSC) Continuous, as soon as the receiver is enabled, and immediately after the
… end of transfer of the previous data.
861 #  define   AT91C_SSC_START_TX                  ((unsigned int) 0x1 <<  8) //
… (SSC) Transmit/Receive start
862 #  define   AT91C_SSC_START_LOW_RF              ((unsigned int) 0x2 <<  8) //
… (SSC) Detection of a low level on RF input
863 #  define   AT91C_SSC_START_HIGH_RF             ((unsigned int) 0x3 <<  8) //
… (SSC) Detection of a high level on RF input
864 #  define   AT91C_SSC_START_FALL_RF             ((unsigned int) 0x4 <<  8) //
… (SSC) Detection of a falling edge on RF input
865 #  define   AT91C_SSC_START_RISE_RF             ((unsigned int) 0x5 <<  8) //
… (SSC) Detection of a rising edge on RF input
866 #  define   AT91C_SSC_START_LEVEL_RF            ((unsigned int) 0x6 <<  8) //
… (SSC) Detection of any level change on RF input
867 #  define   AT91C_SSC_START_EDGE_RF             ((unsigned int) 0x7 <<  8) //
… (SSC) Detection of any edge on RF input
868 #  define   AT91C_SSC_START_0                   ((unsigned int) 0x8 <<  8) //
```

```
868… (SSC) Compare 0
869 #  define AT91C_SSC_STOP        ((unsigned int) 0x1 << 12)  // (SSC) Receive
…   Stop Selection
870 #  define AT91C_SSC_STTOUT      ((unsigned int) 0x1 << 15)  // (SSC)
…   Receive/Transmit Start Output Selection
871 #  define AT91C_SSC_STTDLY      ((unsigned int) 0xFF << 16) // (SSC)
…   Receive/Transmit Start Delay
872 #  define AT91C_SSC_PERIOD      ((unsigned int) 0xFF << 24) // (SSC)
…   Receive/Transmit Period Divider Selection
873 // -------- SSC_RFMR : (SSC Offset: 0x14) SSC Receive Frame Mode Register
…   --------
874 #  define AT91C_SSC_DATLEN      ((unsigned int) 0x1F <<  0) // (SSC) Data
…   Length
875 #  define AT91C_SSC_LOOP        ((unsigned int) 0x1 <<  5)  // (SSC) Loop Mode
876 #  define AT91C_SSC_MSBF        ((unsigned int) 0x1 <<  7)  // (SSC) Most
…   Significant Bit First
877 #  define AT91C_SSC_DATNB       ((unsigned int) 0xF <<  8)  // (SSC) Data
…   Number per Frame
878 #  define AT91C_SSC_FSLEN       ((unsigned int) 0xF << 16)  // (SSC)
…   Receive/Transmit Frame Sync length
879 #  define AT91C_SSC_FSOS        ((unsigned int) 0x7 << 20)  // (SSC)
…   Receive/Transmit Frame Sync Output Selection
880 #  define   AT91C_SSC_FSOS_NONE                 ((unsigned int) 0x0 << 20)  //
…   (SSC) Selected Receive/Transmit Frame Sync Signal: None RK pin Input-only
881 #  define   AT91C_SSC_FSOS_NEGATIVE             ((unsigned int) 0x1 << 20)  //
…   (SSC) Selected Receive/Transmit Frame Sync Signal: Negative Pulse
882 #  define   AT91C_SSC_FSOS_POSITIVE             ((unsigned int) 0x2 << 20)  //
…   (SSC) Selected Receive/Transmit Frame Sync Signal: Positive Pulse
883 #  define   AT91C_SSC_FSOS_LOW                  ((unsigned int) 0x3 << 20)  //
…   (SSC) Selected Receive/Transmit Frame Sync Signal: Driver Low during data
…   transfer
884 #  define   AT91C_SSC_FSOS_HIGH                 ((unsigned int) 0x4 << 20)  //
…   (SSC) Selected Receive/Transmit Frame Sync Signal: Driver High during data
…   transfer
885 #  define   AT91C_SSC_FSOS_TOGGLE               ((unsigned int) 0x5 << 20)  //
…   (SSC) Selected Receive/Transmit Frame Sync Signal: Toggling at each start of
…   data transfer
886 #  define AT91C_SSC_FSEDGE      ((unsigned int) 0x1 << 24)  // (SSC) Frame Sync
…   Edge Detection
887 // -------- SSC_TCMR : (SSC Offset: 0x18) SSC Transmit Clock Mode Register
```

```
887… --------
888 // -------- SSC_TFMR : (SSC Offset: 0x1c) SSC Transmit Frame Mode Register
    … --------
889 #  define AT91C_SSC_DATDEF      ((unsigned int) 0x1 <<  5)  // (SSC) Data
    … Default Value
890 #  define AT91C_SSC_FSDEN       ((unsigned int) 0x1 << 23)  // (SSC) Frame Sync
    … Data Enable
891 // -------- SSC_SR : (SSC Offset: 0x40) SSC Status Register --------
892 #  define AT91C_SSC_TXRDY       ((unsigned int) 0x1 <<  0)  // (SSC) Transmit
    … Ready
893 #  define AT91C_SSC_TXEMPTY     ((unsigned int) 0x1 <<  1)  // (SSC) Transmit
    … Empty
894 #  define AT91C_SSC_ENDTX       ((unsigned int) 0x1 <<  2)  // (SSC) End Of
    … Transmission
895 #  define AT91C_SSC_TXBUFE      ((unsigned int) 0x1 <<  3)  // (SSC) Transmit
    … Buffer Empty
896 #  define AT91C_SSC_RXRDY       ((unsigned int) 0x1 <<  4)  // (SSC) Receive
    … Ready
897 #  define AT91C_SSC_OVRUN       ((unsigned int) 0x1 <<  5)  // (SSC) Receive
    … Overrun
898 #  define AT91C_SSC_ENDRX       ((unsigned int) 0x1 <<  6)  // (SSC) End of
    … Reception
899 #  define AT91C_SSC_RXBUFF      ((unsigned int) 0x1 <<  7)  // (SSC) Receive
    … Buffer Full
900 #  define AT91C_SSC_CP0         ((unsigned int) 0x1 <<  8)  // (SSC) Compare 0
901 #  define AT91C_SSC_CP1         ((unsigned int) 0x1 <<  9)  // (SSC) Compare 1
902 #  define AT91C_SSC_TXSYN       ((unsigned int) 0x1 << 10)  // (SSC) Transmit
    … Sync
903 #  define AT91C_SSC_RXSYN       ((unsigned int) 0x1 << 11)  // (SSC) Receive
    … Sync
904 #  define AT91C_SSC_TXENA       ((unsigned int) 0x1 << 16)  // (SSC) Transmit
    … Enable
905 #  define AT91C_SSC_RXENA       ((unsigned int) 0x1 << 17)  // (SSC) Receive
    … Enable
906 // -------- SSC_IER : (SSC Offset: 0x44) SSC Interrupt Enable Register --------
907 // -------- SSC_IDR : (SSC Offset: 0x48) SSC Interrupt Disable Register
    … --------
908 // -------- SSC_IMR : (SSC Offset: 0x4c) SSC Interrupt Mask Register --------
909
910 //
```

```
910… ********************************************************************************
911  //               SOFTWARE API DEFINITION  FOR Usart
912  //
…    ********************************************************************************
913  typedef struct _AT91S_USART {
914    AT91_REG US_CR;        // Control Register
915    AT91_REG US_MR;        // Mode Register
916    AT91_REG US_IER;       // Interrupt Enable Register
917    AT91_REG US_IDR;       // Interrupt Disable Register
918    AT91_REG US_IMR;       // Interrupt Mask Register
919    AT91_REG US_CSR;       // Channel Status Register
920    AT91_REG US_RHR;       // Receiver Holding Register
921    AT91_REG US_THR;       // Transmitter Holding Register
922    AT91_REG US_BRGR;      // Baud Rate Generator Register
923    AT91_REG US_RTOR;      // Receiver Time-out Register
924    AT91_REG US_TTGR;      // Transmitter Time-guard Register
925    AT91_REG Reserved0[5];    //
926    AT91_REG US_FIDI;      // FI_DI_Ratio Register
927    AT91_REG US_NER;       // Nb Errors Register
928    AT91_REG US_XXR;       // XON_XOFF Register
929    AT91_REG US_IF;        // IRDA_FILTER Register
930    AT91_REG Reserved1[44];   //
931    AT91_REG US_RPR;       // Receive Pointer Register
932    AT91_REG US_RCR;       // Receive Counter Register
933    AT91_REG US_TPR;       // Transmit Pointer Register
934    AT91_REG US_TCR;       // Transmit Counter Register
935    AT91_REG US_RNPR;      // Receive Next Pointer Register
936    AT91_REG US_RNCR;      // Receive Next Counter Register
937    AT91_REG US_TNPR;      // Transmit Next Pointer Register
938    AT91_REG US_TNCR;      // Transmit Next Counter Register
939    AT91_REG US_PTCR;      // PDC Transfer Control Register
940    AT91_REG US_PTSR;      // PDC Transfer Status Register
941  } AT91S_USART, *AT91PS_USART;
942
943  // -------- US_CR : (USART Offset: 0x0) Debug Unit Control Register --------
944  #  define AT91C_US_RSTSTA       ((unsigned int) 0x1 <<  8)  // (USART) Reset
…    Status Bits
945  #  define AT91C_US_STTBRK       ((unsigned int) 0x1 <<  9)  // (USART) Start
…    Break
946  #  define AT91C_US_STPBRK       ((unsigned int) 0x1 << 10)  // (USART) Stop
```

```
946… Break
947  #  define AT91C_US_STTTO        ((unsigned int) 0x1 << 11)  // (USART) Start
…    Time-out
948  #  define AT91C_US_SENDA        ((unsigned int) 0x1 << 12)  // (USART) Send
…    Address
949  #  define AT91C_US_RSTIT        ((unsigned int) 0x1 << 13)  // (USART) Reset
…    Iterations
950  #  define AT91C_US_RSTNACK      ((unsigned int) 0x1 << 14)  // (USART) Reset
…    Non Acknowledge
951  #  define AT91C_US_RETTO        ((unsigned int) 0x1 << 15)  // (USART) Rearm
…    Time-out
952  #  define AT91C_US_DTREN        ((unsigned int) 0x1 << 16)  // (USART) Data
…    Terminal ready Enable
953  #  define AT91C_US_DTRDIS       ((unsigned int) 0x1 << 17)  // (USART) Data
…    Terminal ready Disable
954  #  define AT91C_US_RTSEN        ((unsigned int) 0x1 << 18)  // (USART) Request
…    to Send enable
955  #  define AT91C_US_RTSDIS       ((unsigned int) 0x1 << 19)  // (USART) Request
…    to Send Disable
956  // -------- US_MR : (USART Offset: 0x4) Debug Unit Mode Register --------
957  #  define AT91C_US_USMODE       ((unsigned int) 0xF <<  0)  // (USART) Usart
…    mode
958  #  define   AT91C_US_USMODE_NORMAL              ((unsigned int) 0x0)   //
…    (USART) Normal
959  #  define   AT91C_US_USMODE_RS485               ((unsigned int) 0x1)   //
…    (USART) RS485
960  #  define   AT91C_US_USMODE_HWHSH               ((unsigned int) 0x2)   //
…    (USART) Hardware Handshaking
961  #  define   AT91C_US_USMODE_MODEM               ((unsigned int) 0x3)   //
…    (USART) Modem
962  #  define   AT91C_US_USMODE_ISO7816_0           ((unsigned int) 0x4)   //
…    (USART) ISO7816 protocol: T = 0
963  #  define   AT91C_US_USMODE_ISO7816_1           ((unsigned int) 0x6)   //
…    (USART) ISO7816 protocol: T = 1
964  #  define   AT91C_US_USMODE_IRDA                ((unsigned int) 0x8)   //
…    (USART) IrDA
965  #  define   AT91C_US_USMODE_SWHSH               ((unsigned int) 0xC)   //
…    (USART) Software Handshaking
966  #  define AT91C_US_CLKS         ((unsigned int) 0x3 <<  4)  // (USART) Clock
…    Selection (Baud Rate generator Input Clock
```

```
967 #  define   AT91C_US_CLKS_CLOCK               ((unsigned int) 0x0 <<  4)   //
… (USART) Clock
968 #  define   AT91C_US_CLKS_FDIV1               ((unsigned int) 0x1 <<  4)   //
… (USART) fdiv1
969 #  define   AT91C_US_CLKS_SLOW                ((unsigned int) 0x2 <<  4)   //
… (USART) slow_clock (ARM)
970 #  define   AT91C_US_CLKS_EXT                 ((unsigned int) 0x3 <<  4)   //
… (USART) External (SCK)
971 #  define AT91C_US_CHRL         ((unsigned int) 0x3 <<  6)  // (USART) Clock
… Selection (Baud Rate generator Input Clock
972 #  define   AT91C_US_CHRL_5_BITS              ((unsigned int) 0x0 <<  6)   //
… (USART) Character Length: 5 bits
973 #  define   AT91C_US_CHRL_6_BITS              ((unsigned int) 0x1 <<  6)   //
… (USART) Character Length: 6 bits
974 #  define   AT91C_US_CHRL_7_BITS              ((unsigned int) 0x2 <<  6)   //
… (USART) Character Length: 7 bits
975 #  define   AT91C_US_CHRL_8_BITS              ((unsigned int) 0x3 <<  6)   //
… (USART) Character Length: 8 bits
976 #  define AT91C_US_SYNC         ((unsigned int) 0x1 <<  8)  // (USART)
… Synchronous Mode Select
977 #  define AT91C_US_NBSTOP       ((unsigned int) 0x3 << 12)  // (USART) Number
… of Stop bits
978 #  define   AT91C_US_NBSTOP_1_BIT               ((unsigned int) 0x0 << 12) //
… (USART) 1 stop bit
979 #  define   AT91C_US_NBSTOP_15_BIT              ((unsigned int) 0x1 << 12) //
… (USART) Asynchronous (SYNC=0) 2 stop bits Synchronous (SYNC=1) 2 stop bits
980 #  define   AT91C_US_NBSTOP_2_BIT               ((unsigned int) 0x2 << 12) //
… (USART) 2 stop bits
981 #  define AT91C_US_MSBF         ((unsigned int) 0x1 << 16)  // (USART) Bit
… Order
982 #  define AT91C_US_MODE9        ((unsigned int) 0x1 << 17)  // (USART) 9-bit
… Character length
983 #  define AT91C_US_CKLO         ((unsigned int) 0x1 << 18)  // (USART) Clock
… Output Select
984 #  define AT91C_US_OVER         ((unsigned int) 0x1 << 19)  // (USART) Over
… Sampling Mode
985 #  define AT91C_US_INACK        ((unsigned int) 0x1 << 20)  // (USART) Inhibit
… Non Acknowledge
986 #  define AT91C_US_DSNACK       ((unsigned int) 0x1 << 21)  // (USART) Disable
… Successive NACK
```

```
987 #  define AT91C_US_MAX_ITER     ((unsigned int) 0x1 << 24)  // (USART) Number
    … of Repetitions
988 #  define AT91C_US_FILTER       ((unsigned int) 0x1 << 28)  // (USART) Receive
    … Line Filter
989 // -------- US_IER : (USART Offset: 0x8) Debug Unit Interrupt Enable Register
    … --------
990 #  define AT91C_US_RXBRK        ((unsigned int) 0x1 <<  2)  // (USART) Break
    … Received/End of Break
991 #  define AT91C_US_TIMEOUT      ((unsigned int) 0x1 <<  8)  // (USART) Receiver
    … Time-out
992 #  define AT91C_US_ITERATION    ((unsigned int) 0x1 << 10)  // (USART) Max
    … number of Repetitions Reached
993 #  define AT91C_US_NACK         ((unsigned int) 0x1 << 13)  // (USART) Non
    … Acknowledge
994 #  define AT91C_US_RIIC         ((unsigned int) 0x1 << 16)  // (USART) Ring
    … INdicator Input Change Flag
995 #  define AT91C_US_DSRIC        ((unsigned int) 0x1 << 17)  // (USART) Data Set
    … Ready Input Change Flag
996 #  define AT91C_US_DCDIC        ((unsigned int) 0x1 << 18)  // (USART) Data
    … Carrier Flag
997 #  define AT91C_US_CTSIC        ((unsigned int) 0x1 << 19)  // (USART) Clear To
    … Send Input Change Flag
998 // -------- US_IDR : (USART Offset: 0xc) Debug Unit Interrupt Disable Register
    … --------
999 // -------- US_IMR : (USART Offset: 0x10) Debug Unit Interrupt Mask Register
    … --------
1000 // -------- US_CSR : (USART Offset: 0x14) Debug Unit Channel Status Register
    … --------
1001 #  define AT91C_US_RI           ((unsigned int) 0x1 << 20)  // (USART) Image of
    … RI Input
1002 #  define AT91C_US_DSR          ((unsigned int) 0x1 << 21)  // (USART) Image of
    … DSR Input
1003 #  define AT91C_US_DCD          ((unsigned int) 0x1 << 22)  // (USART) Image of
    … DCD Input
1004 #  define AT91C_US_CTS          ((unsigned int) 0x1 << 23)  // (USART) Image of
    … CTS Input
1005
1006 //
    … ******************************************************************************
1007 //               SOFTWARE API DEFINITION  FOR Two-wire Interface
```

```
1008 //
     ********************************************************************************
1009 typedef struct _AT91S_TWI {
1010   AT91_REG TWI_CR;       // Control Register
1011   AT91_REG TWI_MMR;      // Master Mode Register
1012   AT91_REG TWI_SMR;      // Slave Mode Register
1013   AT91_REG TWI_IADR;         // Internal Address Register
1014   AT91_REG TWI_CWGR;         // Clock Waveform Generator Register
1015   AT91_REG Reserved0[3];     //
1016   AT91_REG TWI_SR;       // Status Register
1017   AT91_REG TWI_IER;      // Interrupt Enable Register
1018   AT91_REG TWI_IDR;      // Interrupt Disable Register
1019   AT91_REG TWI_IMR;      // Interrupt Mask Register
1020   AT91_REG TWI_RHR;      // Receive Holding Register
1021   AT91_REG TWI_THR;      // Transmit Holding Register
1022 } AT91S_TWI, *AT91PS_TWI;
1023
1024 // -------- TWI_CR : (TWI Offset: 0x0) TWI Control Register --------
1025 #  define AT91C_TWI_START       ((unsigned int) 0x1 <<  0)  // (TWI) Send a
     START Condition
1026 #  define AT91C_TWI_STOP        ((unsigned int) 0x1 <<  1)  // (TWI) Send a
     STOP Condition
1027 #  define AT91C_TWI_MSEN        ((unsigned int) 0x1 <<  2)  // (TWI) TWI Master
     Transfer Enabled
1028 #  define AT91C_TWI_MSDIS       ((unsigned int) 0x1 <<  3)  // (TWI) TWI Master
     Transfer Disabled
1029 #  define AT91C_TWI_SVEN        ((unsigned int) 0x1 <<  4)  // (TWI) TWI Slave
     Transfer Enabled
1030 #  define AT91C_TWI_SVDIS       ((unsigned int) 0x1 <<  5)  // (TWI) TWI Slave
     Transfer Disabled
1031 #  define AT91C_TWI_SWRST       ((unsigned int) 0x1 <<  7)  // (TWI) Software
     Reset
1032 // -------- TWI_MMR : (TWI Offset: 0x4) TWI Master Mode Register --------
1033 #  define AT91C_TWI_IADRSZ      ((unsigned int) 0x3 <<  8)  // (TWI) Internal
     Device Address Size
1034 #  define   AT91C_TWI_IADRSZ_NO                 ((unsigned int) 0x0 <<  8)
     // (TWI) No internal device address
1035 #  define   AT91C_TWI_IADRSZ_1_BYTE             ((unsigned int) 0x1 <<  8)
     // (TWI) One-byte internal device address
1036 #  define   AT91C_TWI_IADRSZ_2_BYTE             ((unsigned int) 0x2 <<  8)
```

```
1036… // (TWI) Two-byte internal device address
1037  #  define   AT91C_TWI_IADRSZ_3_BYTE             ((unsigned int) 0x3 <<  8)
…     // (TWI) Three-byte internal device address
1038  #  define AT91C_TWI_MREAD       ((unsigned int) 0x1 << 12)  // (TWI) Master
…     Read Direction
1039  #  define AT91C_TWI_DADR        ((unsigned int) 0x7F << 16) // (TWI) Device
…     Address
1040  // -------- TWI_SMR : (TWI Offset: 0x8) TWI Slave Mode Register --------
1041  #  define AT91C_TWI_SADR        ((unsigned int) 0x7F << 16) // (TWI) Slave
…     Device Address
1042  // -------- TWI_CWGR : (TWI Offset: 0x10) TWI Clock Waveform Generator Register
…     --------
1043  #  define AT91C_TWI_CLDIV       ((unsigned int) 0xFF <<  0) // (TWI) Clock Low
…     Divider
1044  #  define AT91C_TWI_CHDIV       ((unsigned int) 0xFF <<  8) // (TWI) Clock High
…     Divider
1045  #  define AT91C_TWI_CKDIV       ((unsigned int) 0x7 << 16)  // (TWI) Clock
…     Divider
1046  // -------- TWI_SR : (TWI Offset: 0x20) TWI Status Register --------
1047  #  define AT91C_TWI_TXCOMP      ((unsigned int) 0x1 <<  0)  // (TWI)
…     Transmission Completed
1048  #  define AT91C_TWI_RXRDY       ((unsigned int) 0x1 <<  1)  // (TWI) Receive
…     holding register ReaDY
1049  #  define AT91C_TWI_TXRDY       ((unsigned int) 0x1 <<  2)  // (TWI) Transmit
…     holding register ReaDY
1050  #  define AT91C_TWI_SVREAD      ((unsigned int) 0x1 <<  3)  // (TWI) Slave Read
1051  #  define AT91C_TWI_SVACC       ((unsigned int) 0x1 <<  4)  // (TWI) Slave
…     Access
1052  #  define AT91C_TWI_GCACC       ((unsigned int) 0x1 <<  5)  // (TWI) General
…     Call Access
1053  #  define AT91C_TWI_OVRE        ((unsigned int) 0x1 <<  6)  // (TWI) Overrun
…     Error
1054  #  define AT91C_TWI_UNRE        ((unsigned int) 0x1 <<  7)  // (TWI) Underrun
…     Error
1055  #  define AT91C_TWI_NACK        ((unsigned int) 0x1 <<  8)  // (TWI) Not
…     Acknowledged
1056  #  define AT91C_TWI_ARBLST      ((unsigned int) 0x1 <<  9)  // (TWI)
…     Arbitration Lost
1057  // -------- TWI_IER : (TWI Offset: 0x24) TWI Interrupt Enable Register --------
1058  // -------- TWI_IDR : (TWI Offset: 0x28) TWI Interrupt Disable Register
```

```
1058… --------
1059 // -------- TWI_IMR : (TWI Offset: 0x2c) TWI Interrupt Mask Register --------
1060
1061 //
     ******************************************************************************
1062 //               SOFTWARE API DEFINITION  FOR Timer Counter Channel Interface
1063 //
     ******************************************************************************
1064 typedef struct _AT91S_TC {
1065   AT91_REG TC_CCR;      // Channel Control Register
1066   AT91_REG TC_CMR;      // Channel Mode Register (Capture Mode / Waveform Mode)
1067   AT91_REG Reserved0[2];    //
1068   AT91_REG TC_CV;       // Counter Value
1069   AT91_REG TC_RA;       // Register A
1070   AT91_REG TC_RB;       // Register B
1071   AT91_REG TC_RC;       // Register C
1072   AT91_REG TC_SR;       // Status Register
1073   AT91_REG TC_IER;      // Interrupt Enable Register
1074   AT91_REG TC_IDR;      // Interrupt Disable Register
1075   AT91_REG TC_IMR;      // Interrupt Mask Register
1076 } AT91S_TC, *AT91PS_TC;
1077
1078 // -------- TC_CCR : (TC Offset: 0x0) TC Channel Control Register --------
1079 #  define AT91C_TC_CLKEN        ((unsigned int) 0x1 <<  0)  // (TC) Counter
     Clock Enable Command
1080 #  define AT91C_TC_CLKDIS       ((unsigned int) 0x1 <<  1)  // (TC) Counter
     Clock Disable Command
1081 #  define AT91C_TC_SWTRG        ((unsigned int) 0x1 <<  2)  // (TC) Software
     Trigger Command
1082 // -------- TC_CMR : (TC Offset: 0x4) TC Channel Mode Register: Capture Mode /
     Waveform Mode --------
1083 #  define AT91C_TC_CLKS         ((unsigned int) 0x7 <<  0)  // (TC) Clock
     Selection
1084 #  define   AT91C_TC_CLKS_TIMER_DIV1_CLOCK     ((unsigned int) 0x0) // (TC)
     Clock selected: TIMER_DIV1_CLOCK
1085 #  define   AT91C_TC_CLKS_TIMER_DIV2_CLOCK     ((unsigned int) 0x1) // (TC)
     Clock selected: TIMER_DIV2_CLOCK
1086 #  define   AT91C_TC_CLKS_TIMER_DIV3_CLOCK     ((unsigned int) 0x2) // (TC)
     Clock selected: TIMER_DIV3_CLOCK
1087 #  define   AT91C_TC_CLKS_TIMER_DIV4_CLOCK     ((unsigned int) 0x3) // (TC)
```

```
1087… Clock selected: TIMER_DIV4_CLOCK
1088 #  define   AT91C_TC_CLKS_TIMER_DIV5_CLOCK     ((unsigned int) 0x4) // (TC)
   … Clock selected: TIMER_DIV5_CLOCK
1089 #  define   AT91C_TC_CLKS_XC0                  ((unsigned int) 0x5) // (TC)
   … Clock selected: XC0
1090 #  define   AT91C_TC_CLKS_XC1                  ((unsigned int) 0x6) // (TC)
   … Clock selected: XC1
1091 #  define   AT91C_TC_CLKS_XC2                  ((unsigned int) 0x7) // (TC)
   … Clock selected: XC2
1092 #  define AT91C_TC_CLKI         ((unsigned int) 0x1 <<  3)  // (TC) Clock
   … Invert
1093 #  define AT91C_TC_BURST        ((unsigned int) 0x3 <<  4)  // (TC) Burst
   … Signal Selection
1094 #  define   AT91C_TC_BURST_NONE                ((unsigned int) 0x0 <<  4)  //
   … (TC) The clock is not gated by an external signal
1095 #  define   AT91C_TC_BURST_XC0                 ((unsigned int) 0x1 <<  4)  //
   … (TC) XC0 is ANDed with the selected clock
1096 #  define   AT91C_TC_BURST_XC1                 ((unsigned int) 0x2 <<  4)  //
   … (TC) XC1 is ANDed with the selected clock
1097 #  define   AT91C_TC_BURST_XC2                 ((unsigned int) 0x3 <<  4)  //
   … (TC) XC2 is ANDed with the selected clock
1098 #  define AT91C_TC_CPCSTOP      ((unsigned int) 0x1 <<  6)  // (TC) Counter
   … Clock Stopped with RC Compare
1099 #  define AT91C_TC_LDBSTOP      ((unsigned int) 0x1 <<  6)  // (TC) Counter
   … Clock Stopped with RB Loading
1100 #  define AT91C_TC_LDBDIS       ((unsigned int) 0x1 <<  7)  // (TC) Counter
   … Clock Disabled with RB Loading
1101 #  define AT91C_TC_CPCDIS       ((unsigned int) 0x1 <<  7)  // (TC) Counter
   … Clock Disable with RC Compare
1102 #  define AT91C_TC_ETRGEDG      ((unsigned int) 0x3 <<  8)  // (TC) External
   … Trigger Edge Selection
1103 #  define   AT91C_TC_ETRGEDG_NONE                ((unsigned int) 0x0 <<  8)
   … // (TC) Edge: None
1104 #  define   AT91C_TC_ETRGEDG_RISING              ((unsigned int) 0x1 <<  8)
   … // (TC) Edge: rising edge
1105 #  define   AT91C_TC_ETRGEDG_FALLING             ((unsigned int) 0x2 <<  8)
   … // (TC) Edge: falling edge
1106 #  define   AT91C_TC_ETRGEDG_BOTH                ((unsigned int) 0x3 <<  8)
   … // (TC) Edge: each edge
1107 #  define AT91C_TC_EEVTEDG      ((unsigned int) 0x3 <<  8)  // (TC) External
```

```
1107… Event Edge Selection
1108  #  define   AT91C_TC_EEVTEDG_NONE                ((unsigned int) 0x0 <<  8)
…     // (TC) Edge: None
1109  #  define   AT91C_TC_EEVTEDG_RISING              ((unsigned int) 0x1 <<  8)
…     // (TC) Edge: rising edge
1110  #  define   AT91C_TC_EEVTEDG_FALLING             ((unsigned int) 0x2 <<  8)
…     // (TC) Edge: falling edge
1111  #  define   AT91C_TC_EEVTEDG_BOTH                ((unsigned int) 0x3 <<  8)
…     // (TC) Edge: each edge
1112  #  define AT91C_TC_ABETRG       ((unsigned int) 0x1 << 10)  // (TC) TIOA or
…     TIOB External Trigger Selection
1113  #  define AT91C_TC_EEVT         ((unsigned int) 0x3 << 10)  // (TC) External
…     Event  Selection
1114  #  define   AT91C_TC_EEVT_NONE                ((unsigned int) 0x0 << 10)   //
…     (TC) Signal selected as external event: TIOB TIOB direction: input
1115  #  define   AT91C_TC_EEVT_RISING              ((unsigned int) 0x1 << 10)   //
…     (TC) Signal selected as external event: XC0 TIOB direction: output
1116  #  define   AT91C_TC_EEVT_FALLING             ((unsigned int) 0x2 << 10)   //
…     (TC) Signal selected as external event: XC1 TIOB direction: output
1117  #  define   AT91C_TC_EEVT_BOTH                ((unsigned int) 0x3 << 10)   //
…     (TC) Signal selected as external event: XC2 TIOB direction: output
1118  #  define AT91C_TC_ENETRG       ((unsigned int) 0x1 << 12)  // (TC) External
…     Event Trigger enable
1119  #  define AT91C_TC_WAVESEL      ((unsigned int) 0x3 << 13)  // (TC) Waveform
…     Selection
1120  #  define   AT91C_TC_WAVESEL_UP                ((unsigned int) 0x0 << 13)
…     // (TC) UP mode without atomatic trigger on RC Compare
1121  #  define   AT91C_TC_WAVESEL_UPDOWN            ((unsigned int) 0x1 << 13)
…     // (TC) UPDOWN mode without automatic trigger on RC Compare
1122  #  define   AT91C_TC_WAVESEL_UP_AUTO           ((unsigned int) 0x2 << 13)
…     // (TC) UP mode with automatic trigger on RC Compare
1123  #  define   AT91C_TC_WAVESEL_UPDOWN_AUTO       ((unsigned int) 0x3 << 13)
…     // (TC) UPDOWN mode with automatic trigger on RC Compare
1124  #  define AT91C_TC_CPCTRG       ((unsigned int) 0x1 << 14)  // (TC) RC Compare
…     Trigger Enable
1125  #  define AT91C_TC_WAVE         ((unsigned int) 0x1 << 15)  // (TC)
1126  #  define AT91C_TC_LDRA         ((unsigned int) 0x3 << 16)  // (TC) RA Loading
…     Selection
1127  #  define   AT91C_TC_LDRA_NONE                ((unsigned int) 0x0 << 16)   //
…     (TC) Edge: None
```

```
1128 #  define   AT91C_TC_LDRA_RISING             ((unsigned int) 0x1 << 16)   //
…  (TC) Edge: rising edge of TIOA
1129 #  define   AT91C_TC_LDRA_FALLING            ((unsigned int) 0x2 << 16)   //
…  (TC) Edge: falling edge of TIOA
1130 #  define   AT91C_TC_LDRA_BOTH               ((unsigned int) 0x3 << 16)   //
…  (TC) Edge: each edge of TIOA
1131 #  define AT91C_TC_ACPA         ((unsigned int) 0x3 << 16)  // (TC) RA Compare
…  Effect on TIOA
1132 #  define   AT91C_TC_ACPA_NONE               ((unsigned int) 0x0 << 16)   //
…  (TC) Effect: none
1133 #  define   AT91C_TC_ACPA_SET                ((unsigned int) 0x1 << 16)   //
…  (TC) Effect: set
1134 #  define   AT91C_TC_ACPA_CLEAR              ((unsigned int) 0x2 << 16)   //
…  (TC) Effect: clear
1135 #  define   AT91C_TC_ACPA_TOGGLE             ((unsigned int) 0x3 << 16)   //
…  (TC) Effect: toggle
1136 #  define AT91C_TC_LDRB         ((unsigned int) 0x3 << 18)  // (TC) RB Loading
…  Selection
1137 #  define   AT91C_TC_LDRB_NONE               ((unsigned int) 0x0 << 18)   //
…  (TC) Edge: None
1138 #  define   AT91C_TC_LDRB_RISING             ((unsigned int) 0x1 << 18)   //
…  (TC) Edge: rising edge of TIOA
1139 #  define   AT91C_TC_LDRB_FALLING            ((unsigned int) 0x2 << 18)   //
…  (TC) Edge: falling edge of TIOA
1140 #  define   AT91C_TC_LDRB_BOTH               ((unsigned int) 0x3 << 18)   //
…  (TC) Edge: each edge of TIOA
1141 #  define AT91C_TC_ACPC         ((unsigned int) 0x3 << 18)  // (TC) RC Compare
…  Effect on TIOA
1142 #  define   AT91C_TC_ACPC_NONE               ((unsigned int) 0x0 << 18)   //
…  (TC) Effect: none
1143 #  define   AT91C_TC_ACPC_SET                ((unsigned int) 0x1 << 18)   //
…  (TC) Effect: set
1144 #  define   AT91C_TC_ACPC_CLEAR              ((unsigned int) 0x2 << 18)   //
…  (TC) Effect: clear
1145 #  define   AT91C_TC_ACPC_TOGGLE             ((unsigned int) 0x3 << 18)   //
…  (TC) Effect: toggle
1146 #  define AT91C_TC_AEEVT        ((unsigned int) 0x3 << 20)  // (TC) External
…  Event Effect on TIOA
1147 #  define   AT91C_TC_AEEVT_NONE               ((unsigned int) 0x0 << 20)  //
…  (TC) Effect: none
```

```
1148 #  define   AT91C_TC_AEEVT_SET                ((unsigned int) 0x1 << 20)  //
…    (TC) Effect: set
1149 #  define   AT91C_TC_AEEVT_CLEAR              ((unsigned int) 0x2 << 20)  //
…    (TC) Effect: clear
1150 #  define   AT91C_TC_AEEVT_TOGGLE             ((unsigned int) 0x3 << 20)  //
…    (TC) Effect: toggle
1151 #  define AT91C_TC_ASWTRG       ((unsigned int) 0x3 << 22)  // (TC) Software
…    Trigger Effect on TIOA
1152 #  define   AT91C_TC_ASWTRG_NONE               ((unsigned int) 0x0 << 22) //
…    (TC) Effect: none
1153 #  define   AT91C_TC_ASWTRG_SET                ((unsigned int) 0x1 << 22) //
…    (TC) Effect: set
1154 #  define   AT91C_TC_ASWTRG_CLEAR              ((unsigned int) 0x2 << 22) //
…    (TC) Effect: clear
1155 #  define   AT91C_TC_ASWTRG_TOGGLE             ((unsigned int) 0x3 << 22) //
…    (TC) Effect: toggle
1156 #  define AT91C_TC_BCPB         ((unsigned int) 0x3 << 24)  // (TC) RB Compare
…    Effect on TIOB
1157 #  define   AT91C_TC_BCPB_NONE                ((unsigned int) 0x0 << 24)   //
…    (TC) Effect: none
1158 #  define   AT91C_TC_BCPB_SET                 ((unsigned int) 0x1 << 24)   //
…    (TC) Effect: set
1159 #  define   AT91C_TC_BCPB_CLEAR               ((unsigned int) 0x2 << 24)   //
…    (TC) Effect: clear
1160 #  define   AT91C_TC_BCPB_TOGGLE              ((unsigned int) 0x3 << 24)   //
…    (TC) Effect: toggle
1161 #  define AT91C_TC_BCPC         ((unsigned int) 0x3 << 26)  // (TC) RC Compare
…    Effect on TIOB
1162 #  define   AT91C_TC_BCPC_NONE                ((unsigned int) 0x0 << 26)   //
…    (TC) Effect: none
1163 #  define   AT91C_TC_BCPC_SET                 ((unsigned int) 0x1 << 26)   //
…    (TC) Effect: set
1164 #  define   AT91C_TC_BCPC_CLEAR               ((unsigned int) 0x2 << 26)   //
…    (TC) Effect: clear
1165 #  define   AT91C_TC_BCPC_TOGGLE              ((unsigned int) 0x3 << 26)   //
…    (TC) Effect: toggle
1166 #  define AT91C_TC_BEEVT        ((unsigned int) 0x3 << 28)  // (TC) External
…    Event Effect on TIOB
1167 #  define   AT91C_TC_BEEVT_NONE                ((unsigned int) 0x0 << 28)  //
…    (TC) Effect: none
```

```
1168 #  define   AT91C_TC_BEEVT_SET                  ((unsigned int) 0x1 << 28)  //
   … (TC) Effect: set
1169 #  define   AT91C_TC_BEEVT_CLEAR                ((unsigned int) 0x2 << 28)  //
   … (TC) Effect: clear
1170 #  define   AT91C_TC_BEEVT_TOGGLE               ((unsigned int) 0x3 << 28)  //
   … (TC) Effect: toggle
1171 #  define AT91C_TC_BSWTRG       ((unsigned int) 0x3 << 30)  // (TC) Software
   … Trigger Effect on TIOB
1172 #  define   AT91C_TC_BSWTRG_NONE                 ((unsigned int) 0x0 << 30) //
   … (TC) Effect: none
1173 #  define   AT91C_TC_BSWTRG_SET                  ((unsigned int) 0x1 << 30) //
   … (TC) Effect: set
1174 #  define   AT91C_TC_BSWTRG_CLEAR                ((unsigned int) 0x2 << 30) //
   … (TC) Effect: clear
1175 #  define   AT91C_TC_BSWTRG_TOGGLE               ((unsigned int) 0x3 << 30) //
   … (TC) Effect: toggle
1176 // -------- TC_SR : (TC Offset: 0x20) TC Channel Status Register --------
1177 #  define AT91C_TC_COVFS        ((unsigned int) 0x1 <<  0)  // (TC) Counter
   … Overflow
1178 #  define AT91C_TC_LOVRS        ((unsigned int) 0x1 <<  1)  // (TC) Load
   … Overrun
1179 #  define AT91C_TC_CPAS         ((unsigned int) 0x1 <<  2)  // (TC) RA Compare
1180 #  define AT91C_TC_CPBS         ((unsigned int) 0x1 <<  3)  // (TC) RB Compare
1181 #  define AT91C_TC_CPCS         ((unsigned int) 0x1 <<  4)  // (TC) RC Compare
1182 #  define AT91C_TC_LDRAS        ((unsigned int) 0x1 <<  5)  // (TC) RA Loading
1183 #  define AT91C_TC_LDRBS        ((unsigned int) 0x1 <<  6)  // (TC) RB Loading
1184 #  define AT91C_TC_ETRCS        ((unsigned int) 0x1 <<  7)  // (TC) External
   … Trigger
1185 #  define AT91C_TC_ETRGS        ((unsigned int) 0x1 << 16)  // (TC) Clock
   … Enabling
1186 #  define AT91C_TC_MTIOA        ((unsigned int) 0x1 << 17)  // (TC) TIOA Mirror
1187 #  define AT91C_TC_MTIOB        ((unsigned int) 0x1 << 18)  // (TC) TIOA Mirror
1188 // -------- TC_IER : (TC Offset: 0x24) TC Channel Interrupt Enable Register
   … --------
1189 // -------- TC_IDR : (TC Offset: 0x28) TC Channel Interrupt Disable Register
   … --------
1190 // -------- TC_IMR : (TC Offset: 0x2c) TC Channel Interrupt Mask Register
   … --------
1191
1192 //
```

```
1192… ********************************************************************************
1193  //               SOFTWARE API DEFINITION  FOR Timer Counter Interface
1194  //
…     ********************************************************************************
1195  typedef struct _AT91S_TCB {
1196    AT91S_TC TCB_TC0;     // TC Channel 0
1197    AT91_REG Reserved0[4];    //
1198    AT91S_TC TCB_TC1;     // TC Channel 1
1199    AT91_REG Reserved1[4];    //
1200    AT91S_TC TCB_TC2;     // TC Channel 2
1201    AT91_REG Reserved2[4];    //
1202    AT91_REG TCB_BCR;     // TC Block Control Register
1203    AT91_REG TCB_BMR;     // TC Block Mode Register
1204  } AT91S_TCB, *AT91PS_TCB;
1205
1206  // -------- TCB_BCR : (TCB Offset: 0xc0) TC Block Control Register --------
1207  #  define AT91C_TCB_SYNC        ((unsigned int) 0x1 <<  0)  // (TCB) Synchro
…     Command
1208  // -------- TCB_BMR : (TCB Offset: 0xc4) TC Block Mode Register --------
1209  #  define AT91C_TCB_TC0XC0S     ((unsigned int) 0x1 <<  0)  // (TCB) External
…     Clock Signal 0 Selection
1210  #  define   AT91C_TCB_TC0XC0S_TCLK0               ((unsigned int) 0x0) //
…     (TCB) TCLK0 connected to XC0
1211  #  define   AT91C_TCB_TC0XC0S_NONE                ((unsigned int) 0x1) //
…     (TCB) None signal connected to XC0
1212  #  define   AT91C_TCB_TC0XC0S_TIOA1               ((unsigned int) 0x2) //
…     (TCB) TIOA1 connected to XC0
1213  #  define   AT91C_TCB_TC0XC0S_TIOA2               ((unsigned int) 0x3) //
…     (TCB) TIOA2 connected to XC0
1214  #  define AT91C_TCB_TC1XC1S     ((unsigned int) 0x1 <<  2)  // (TCB) External
…     Clock Signal 1 Selection
1215  #  define   AT91C_TCB_TC1XC1S_TCLK1               ((unsigned int) 0x0 <<  2)
…     // (TCB) TCLK1 connected to XC1
1216  #  define   AT91C_TCB_TC1XC1S_NONE                ((unsigned int) 0x1 <<  2)
…     // (TCB) None signal connected to XC1
1217  #  define   AT91C_TCB_TC1XC1S_TIOA0               ((unsigned int) 0x2 <<  2)
…     // (TCB) TIOA0 connected to XC1
1218  #  define   AT91C_TCB_TC1XC1S_TIOA2               ((unsigned int) 0x3 <<  2)
…     // (TCB) TIOA2 connected to XC1
1219  #  define AT91C_TCB_TC2XC2S     ((unsigned int) 0x1 <<  4)  // (TCB) External
```

```
1219… Clock Signal 2 Selection
1220  #  define   AT91C_TCB_TC2XC2S_TCLK2              ((unsigned int) 0x0 <<  4)
…     // (TCB) TCLK2 connected to XC2
1221  #  define   AT91C_TCB_TC2XC2S_NONE               ((unsigned int) 0x1 <<  4)
…     // (TCB) None signal connected to XC2
1222  #  define   AT91C_TCB_TC2XC2S_TIOA0              ((unsigned int) 0x2 <<  4)
…     // (TCB) TIOA0 connected to XC2
1223  #  define   AT91C_TCB_TC2XC2S_TIOA2              ((unsigned int) 0x3 <<  4)
…     // (TCB) TIOA2 connected to XC2
1224
1225  //
…     ********************************************************************************
1226  //               SOFTWARE API DEFINITION  FOR PWMC Channel Interface
1227  //
…     ********************************************************************************
1228  typedef struct _AT91S_PWMC_CH {
1229    AT91_REG PWMC_CMR;        // Channel Mode Register
1230    AT91_REG PWMC_CDTYR;      // Channel Duty Cycle Register
1231    AT91_REG PWMC_CPRDR;      // Channel Period Register
1232    AT91_REG PWMC_CCNTR;      // Channel Counter Register
1233    AT91_REG PWMC_CUPDR;      // Channel Update Register
1234    AT91_REG PWMC_Reserved[3];    // Reserved
1235  } AT91S_PWMC_CH, *AT91PS_PWMC_CH;
1236
1237  // -------- PWMC_CMR : (PWMC_CH Offset: 0x0) PWMC Channel Mode Register
…     --------
1238  #  define AT91C_PWMC_CPRE       ((unsigned int) 0xF <<  0)  // (PWMC_CH)
…     Channel Pre-scaler : PWMC_CLKx
1239  #  define   AT91C_PWMC_CPRE_MCK                 ((unsigned int) 0x0)   //
…     (PWMC_CH)
1240  #  define   AT91C_PWMC_CPRE_MCKA                ((unsigned int) 0xB)   //
…     (PWMC_CH)
1241  #  define   AT91C_PWMC_CPRE_MCKB                ((unsigned int) 0xC)   //
…     (PWMC_CH)
1242  #  define AT91C_PWMC_CALG       ((unsigned int) 0x1 <<  8)  // (PWMC_CH)
…     Channel Alignment
1243  #  define AT91C_PWMC_CPOL       ((unsigned int) 0x1 <<  9)  // (PWMC_CH)
…     Channel Polarity
1244  #  define AT91C_PWMC_CPD        ((unsigned int) 0x1 << 10)  // (PWMC_CH)
…     Channel Update Period
```

```
1245 // -------- PWMC_CDTYR : (PWMC_CH Offset: 0x4) PWMC Channel Duty Cycle Register
     --------
1246 #  define AT91C_PWMC_CDTY       ((unsigned int) 0x0 <<  0)  // (PWMC_CH)
     Channel Duty Cycle
1247 // -------- PWMC_CPRDR : (PWMC_CH Offset: 0x8) PWMC Channel Period Register
     --------
1248 #  define AT91C_PWMC_CPRD       ((unsigned int) 0x0 <<  0)  // (PWMC_CH)
     Channel Period
1249 // -------- PWMC_CCNTR : (PWMC_CH Offset: 0xc) PWMC Channel Counter Register
     --------
1250 #  define AT91C_PWMC_CCNT       ((unsigned int) 0x0 <<  0)  // (PWMC_CH)
     Channel Counter
1251 // -------- PWMC_CUPDR : (PWMC_CH Offset: 0x10) PWMC Channel Update Register
     --------
1252 #  define AT91C_PWMC_CUPD       ((unsigned int) 0x0 <<  0)  // (PWMC_CH)
     Channel Update
1253
1254 //
     ////////////////////////////////////////////////////////////////////////////
1255 //              SOFTWARE API DEFINITION  FOR Pulse Width Modulation Controller
     Interface
1256 //
     ////////////////////////////////////////////////////////////////////////////
1257 typedef struct _AT91S_PWMC {
1258   AT91_REG PWMC_MR;     // PWMC Mode Register
1259   AT91_REG PWMC_ENA;        // PWMC Enable Register
1260   AT91_REG PWMC_DIS;        // PWMC Disable Register
1261   AT91_REG PWMC_SR;     // PWMC Status Register
1262   AT91_REG PWMC_IER;        // PWMC Interrupt Enable Register
1263   AT91_REG PWMC_IDR;        // PWMC Interrupt Disable Register
1264   AT91_REG PWMC_IMR;        // PWMC Interrupt Mask Register
1265   AT91_REG PWMC_ISR;        // PWMC Interrupt Status Register
1266   AT91_REG Reserved0[55];   //
1267   AT91_REG PWMC_VR;     // PWMC Version Register
1268   AT91_REG Reserved1[64];   //
1269   AT91S_PWMC_CH PWMC_CH[32];    // PWMC Channel 0
1270 } AT91S_PWMC, *AT91PS_PWMC;
1271
1272 // -------- PWMC_MR : (PWMC Offset: 0x0) PWMC Mode Register --------
1273 #  define AT91C_PWMC_DIVA       ((unsigned int) 0xFF <<  0) // (PWMC) CLKA
```

```
1273… divide factor.
1274  #  define AT91C_PWMC_PREA       ((unsigned int) 0xF <<  8)  // (PWMC) Divider
…     Input Clock Prescaler A
1275  #  define   AT91C_PWMC_PREA_MCK                 ((unsigned int) 0x0 <<  8) //
…     (PWMC)
1276  #  define AT91C_PWMC_DIVB       ((unsigned int) 0xFF << 16) // (PWMC) CLKB
…     divide factor.
1277  #  define AT91C_PWMC_PREB       ((unsigned int) 0xF << 24)  // (PWMC) Divider
…     Input Clock Prescaler B
1278  #  define   AT91C_PWMC_PREB_MCK                 ((unsigned int) 0x0 << 24) //
…     (PWMC)
1279  // -------- PWMC_ENA : (PWMC Offset: 0x4) PWMC Enable Register --------
1280  #  define AT91C_PWMC_CHID0      ((unsigned int) 0x1 <<  0)  // (PWMC) Channel
…     ID 0
1281  #  define AT91C_PWMC_CHID1      ((unsigned int) 0x1 <<  1)  // (PWMC) Channel
…     ID 1
1282  #  define AT91C_PWMC_CHID2      ((unsigned int) 0x1 <<  2)  // (PWMC) Channel
…     ID 2
1283  #  define AT91C_PWMC_CHID3      ((unsigned int) 0x1 <<  3)  // (PWMC) Channel
…     ID 3
1284  #  define AT91C_PWMC_CHID4      ((unsigned int) 0x1 <<  4)  // (PWMC) Channel
…     ID 4
1285  #  define AT91C_PWMC_CHID5      ((unsigned int) 0x1 <<  5)  // (PWMC) Channel
…     ID 5
1286  #  define AT91C_PWMC_CHID6      ((unsigned int) 0x1 <<  6)  // (PWMC) Channel
…     ID 6
1287  #  define AT91C_PWMC_CHID7      ((unsigned int) 0x1 <<  7)  // (PWMC) Channel
…     ID 7
1288  // -------- PWMC_DIS : (PWMC Offset: 0x8) PWMC Disable Register --------
1289  // -------- PWMC_SR : (PWMC Offset: 0xc) PWMC Status Register --------
1290  // -------- PWMC_IER : (PWMC Offset: 0x10) PWMC Interrupt Enable Register
…     --------
1291  // -------- PWMC_IDR : (PWMC Offset: 0x14) PWMC Interrupt Disable Register
…     --------
1292  // -------- PWMC_IMR : (PWMC Offset: 0x18) PWMC Interrupt Mask Register
…     --------
1293  // -------- PWMC_ISR : (PWMC Offset: 0x1c) PWMC Interrupt Status Register
…     --------
1294
1295  //
```

```
1295… ********************************************************************************
1296  //               SOFTWARE API DEFINITION  FOR USB Device Interface
1297  //
…     ********************************************************************************
1298  typedef struct _AT91S_UDP {
1299    AT91_REG UDP_NUM;     // Frame Number Register
1300    AT91_REG UDP_GLBSTATE;    // Global State Register
1301    AT91_REG UDP_FADDR;       // Function Address Register
1302    AT91_REG Reserved0[1];    //
1303    AT91_REG UDP_IER;     // Interrupt Enable Register
1304    AT91_REG UDP_IDR;     // Interrupt Disable Register
1305    AT91_REG UDP_IMR;     // Interrupt Mask Register
1306    AT91_REG UDP_ISR;     // Interrupt Status Register
1307    AT91_REG UDP_ICR;     // Interrupt Clear Register
1308    AT91_REG Reserved1[1];    //
1309    AT91_REG UDP_RSTEP;       // Reset Endpoint Register
1310    AT91_REG Reserved2[1];    //
1311    AT91_REG UDP_CSR[8];      // Endpoint Control and Status Register
1312    AT91_REG UDP_FDR[8];      // Endpoint FIFO Data Register
1313  } AT91S_UDP, *AT91PS_UDP;
1314
1315  // -------- UDP_FRM_NUM : (UDP Offset: 0x0) USB Frame Number Register --------
1316  #  define AT91C_UDP_FRM_NUM     ((unsigned int) 0x7FF <<  0)    // (UDP) Frame
…     Number as Defined in the Packet Field Formats
1317  #  define AT91C_UDP_FRM_ERR     ((unsigned int) 0x1 << 16)  // (UDP) Frame
…     Error
1318  #  define AT91C_UDP_FRM_OK      ((unsigned int) 0x1 << 17)  // (UDP) Frame OK
1319  // -------- UDP_GLB_STATE : (UDP Offset: 0x4) USB Global State Register
…     --------
1320  #  define AT91C_UDP_FADDEN      ((unsigned int) 0x1 <<  0)  // (UDP) Function
…     Address Enable
1321  #  define AT91C_UDP_CONFG       ((unsigned int) 0x1 <<  1)  // (UDP) Configured
1322  #  define AT91C_UDP_RMWUPE      ((unsigned int) 0x1 <<  2)  // (UDP) Remote
…     Wake Up Enable
1323  #  define AT91C_UDP_RSMINPR     ((unsigned int) 0x1 <<  3)  // (UDP) A Resume
…     Has Been Sent to the Host
1324  // -------- UDP_FADDR : (UDP Offset: 0x8) USB Function Address Register
…     --------
1325  #  define AT91C_UDP_FADD        ((unsigned int) 0xFF <<  0) // (UDP) Function
…     Address Value
```

```
1326 #  define AT91C_UDP_FEN         ((unsigned int) 0x1 <<  8)  // (UDP) Function
     … Enable
1327 // -------- UDP_IER : (UDP Offset: 0x10) USB Interrupt Enable Register --------
1328 #  define AT91C_UDP_EPINT0      ((unsigned int) 0x1 <<  0)  // (UDP) Endpoint 0
     … Interrupt
1329 #  define AT91C_UDP_EPINT1      ((unsigned int) 0x1 <<  1)  // (UDP) Endpoint 0
     … Interrupt
1330 #  define AT91C_UDP_EPINT2      ((unsigned int) 0x1 <<  2)  // (UDP) Endpoint 2
     … Interrupt
1331 #  define AT91C_UDP_EPINT3      ((unsigned int) 0x1 <<  3)  // (UDP) Endpoint 3
     … Interrupt
1332 #  define AT91C_UDP_EPINT4      ((unsigned int) 0x1 <<  4)  // (UDP) Endpoint 4
     … Interrupt
1333 #  define AT91C_UDP_EPINT5      ((unsigned int) 0x1 <<  5)  // (UDP) Endpoint 5
     … Interrupt
1334 #  define AT91C_UDP_EPINT6      ((unsigned int) 0x1 <<  6)  // (UDP) Endpoint 6
     … Interrupt
1335 #  define AT91C_UDP_EPINT7      ((unsigned int) 0x1 <<  7)  // (UDP) Endpoint 7
     … Interrupt
1336 #  define AT91C_UDP_RXSUSP      ((unsigned int) 0x1 <<  8)  // (UDP) USB
     … Suspend Interrupt
1337 #  define AT91C_UDP_RXRSM       ((unsigned int) 0x1 <<  9)  // (UDP) USB Resume
     … Interrupt
1338 #  define AT91C_UDP_EXTRSM      ((unsigned int) 0x1 << 10)  // (UDP) USB
     … External Resume Interrupt
1339 #  define AT91C_UDP_SOFINT      ((unsigned int) 0x1 << 11)  // (UDP) USB Start
     … Of frame Interrupt
1340 #  define AT91C_UDP_WAKEUP      ((unsigned int) 0x1 << 13)  // (UDP) USB Resume
     … Interrupt
1341 // -------- UDP_IDR : (UDP Offset: 0x14) USB Interrupt Disable Register
     … --------
1342 // -------- UDP_IMR : (UDP Offset: 0x18) USB Interrupt Mask Register --------
1343 // -------- UDP_ISR : (UDP Offset: 0x1c) USB Interrupt Status Register --------
1344 #  define AT91C_UDP_ENDBUSRES   ((unsigned int) 0x1 << 12)  // (UDP) USB End Of
     … Bus Reset Interrupt
1345 // -------- UDP_ICR : (UDP Offset: 0x20) USB Interrupt Clear Register --------
1346 // -------- UDP_RST_EP : (UDP Offset: 0x28) USB Reset Endpoint Register
     … --------
1347 #  define AT91C_UDP_EP0         ((unsigned int) 0x1 <<  0)  // (UDP) Reset
     … Endpoint 0
```

```
1348 #  define AT91C_UDP_EP1         ((unsigned int) 0x1 <<  1)  // (UDP) Reset
   … Endpoint 1
1349 #  define AT91C_UDP_EP2         ((unsigned int) 0x1 <<  2)  // (UDP) Reset
   … Endpoint 2
1350 #  define AT91C_UDP_EP3         ((unsigned int) 0x1 <<  3)  // (UDP) Reset
   … Endpoint 3
1351 #  define AT91C_UDP_EP4         ((unsigned int) 0x1 <<  4)  // (UDP) Reset
   … Endpoint 4
1352 #  define AT91C_UDP_EP5         ((unsigned int) 0x1 <<  5)  // (UDP) Reset
   … Endpoint 5
1353 #  define AT91C_UDP_EP6         ((unsigned int) 0x1 <<  6)  // (UDP) Reset
   … Endpoint 6
1354 #  define AT91C_UDP_EP7         ((unsigned int) 0x1 <<  7)  // (UDP) Reset
   … Endpoint 7
1355 // -------- UDP_CSR : (UDP Offset: 0x30) USB Endpoint Control and Status
   … Register --------
1356 #  define AT91C_UDP_TXCOMP      ((unsigned int) 0x1 <<  0)  // (UDP) Generates
   … an IN packet with data previously written in the DPR
1357 #  define AT91C_UDP_RX_DATA_BK0 ((unsigned int) 0x1 <<  1)  // (UDP) Receive
   … Data Bank 0
1358 #  define AT91C_UDP_RXSETUP     ((unsigned int) 0x1 <<  2)  // (UDP) Sends
   … STALL to the Host (Control endpoints)
1359 #  define AT91C_UDP_ISOERROR    ((unsigned int) 0x1 <<  3)  // (UDP)
   … Isochronous error (Isochronous endpoints)
1360 #  define AT91C_UDP_TXPKTRDY    ((unsigned int) 0x1 <<  4)  // (UDP) Transmit
   … Packet Ready
1361 #  define AT91C_UDP_FORCESTALL  ((unsigned int) 0x1 <<  5)  // (UDP) Force
   … Stall (used by Control, Bulk and Isochronous endpoints).
1362 #  define AT91C_UDP_RX_DATA_BK1 ((unsigned int) 0x1 <<  6)  // (UDP) Receive
   … Data Bank 1 (only used by endpoints with ping-pong attributes).
1363 #  define AT91C_UDP_DIR         ((unsigned int) 0x1 <<  7)  // (UDP) Transfer
   … Direction
1364 #  define AT91C_UDP_EPTYPE      ((unsigned int) 0x7 <<  8)  // (UDP) Endpoint
   … type
1365 #  define   AT91C_UDP_EPTYPE_CTRL                 ((unsigned int) 0x0 <<  8)
   … // (UDP) Control
1366 #  define   AT91C_UDP_EPTYPE_ISO_OUT              ((unsigned int) 0x1 <<  8)
   … // (UDP) Isochronous OUT
1367 #  define   AT91C_UDP_EPTYPE_BULK_OUT             ((unsigned int) 0x2 <<  8)
   … // (UDP) Bulk OUT
```

```
1368 #  define   AT91C_UDP_EPTYPE_INT_OUT             ((unsigned int) 0x3 <<  8)
     … // (UDP) Interrupt OUT
1369 #  define   AT91C_UDP_EPTYPE_ISO_IN              ((unsigned int) 0x5 <<  8)
     … // (UDP) Isochronous IN
1370 #  define   AT91C_UDP_EPTYPE_BULK_IN             ((unsigned int) 0x6 <<  8)
     … // (UDP) Bulk IN
1371 #  define   AT91C_UDP_EPTYPE_INT_IN              ((unsigned int) 0x7 <<  8)
     … // (UDP) Interrupt IN
1372 #  define AT91C_UDP_DTGLE       ((unsigned int) 0x1 << 11)  // (UDP) Data
     … Toggle
1373 #  define AT91C_UDP_EPEDS       ((unsigned int) 0x1 << 15)  // (UDP) Endpoint
     … Enable Disable
1374 #  define AT91C_UDP_RXBYTECNT   ((unsigned int) 0x7FF << 16)    // (UDP) Number
     … Of Bytes Available in the FIFO
1375
1376 //
     … ********************************************************************************
1377 //               REGISTER ADDRESS DEFINITION FOR AT91SAM7S64
1378 //
     … ********************************************************************************
1379 // ========== Register definition for SYSC peripheral ==========
1380 #  define AT91C_SYSC_SYSC_VREG ((AT91_REG *)    0xFFFFFD60) // (SYSC) Voltage
     … Regulator Mode Register
1381 // ========== Register definition for AIC peripheral ==========
1382 #  define AT91C_AIC_ICCR  ((AT91_REG *)     0xFFFFF128)
1383                             // (AIC) Interrupt Clear Command Register
1384 #  define AT91C_AIC_IECR  ((AT91_REG *)     0xFFFFF120)
1385                             // (AIC) Interrupt Enable Command Register
1386 #  define AT91C_AIC_SMR   ((AT91_REG *)     0xFFFFF000)
1387                             // (AIC) Source Mode Register
1388 #  define AT91C_AIC_ISCR  ((AT91_REG *)     0xFFFFF12C)
1389                             // (AIC) Interrupt Set Command Register
1390 #  define AT91C_AIC_EOICR ((AT91_REG *)     0xFFFFF130)
1391                             // (AIC) End of Interrupt Command Register
1392 #  define AT91C_AIC_DCR   ((AT91_REG *)     0xFFFFF138)
1393                             // (AIC) Debug Control Register (Protect)
1394 #  define AT91C_AIC_FFER  ((AT91_REG *)     0xFFFFF140)
1395                             // (AIC) Fast Forcing Enable Register
1396 #  define AT91C_AIC_SVR   ((AT91_REG *)     0xFFFFF080)
1397                             // (AIC) Source Vector Register
```

```
1398 #  define AT91C_AIC_SPU   ((AT91_REG *)     0xFFFFF134)
1399                               // (AIC) Spurious Vector Register
1400 #  define AT91C_AIC_FFDR  ((AT91_REG *)     0xFFFFF144)
1401                               // (AIC) Fast Forcing Disable Register
1402 #  define AT91C_AIC_FVR   ((AT91_REG *)     0xFFFFF104)
1403                               // (AIC) FIQ Vector Register
1404 #  define AT91C_AIC_FFSR  ((AT91_REG *)     0xFFFFF148)
1405                               // (AIC) Fast Forcing Status Register
1406 #  define AT91C_AIC_IMR   ((AT91_REG *)     0xFFFFF110)
1407                               // (AIC) Interrupt Mask Register
1408 #  define AT91C_AIC_ISR   ((AT91_REG *)     0xFFFFF108)
1409                               // (AIC) Interrupt Status Register
1410 #  define AT91C_AIC_IVR   ((AT91_REG *)     0xFFFFF100)
1411                               // (AIC) IRQ Vector Register
1412 #  define AT91C_AIC_IDCR  ((AT91_REG *)     0xFFFFF124)
1413                               // (AIC) Interrupt Disable Command Register
1414 #  define AT91C_AIC_CISR  ((AT91_REG *)     0xFFFFF114)
1415                               // (AIC) Core Interrupt Status Register
1416 #  define AT91C_AIC_IPR   ((AT91_REG *)     0xFFFFF10C)
1417                               // (AIC) Interrupt Pending Register
1418 // ========== Register definition for DBGU peripheral ==========
1419 #  define AT91C_DBGU_C2R  ((AT91_REG *)     0xFFFFF244)
1420                               // (DBGU) Chip ID2 Register
1421 #  define AT91C_DBGU_THR  ((AT91_REG *)     0xFFFFF21C)
1422                               // (DBGU) Transmitter Holding Register
1423 #  define AT91C_DBGU_CSR  ((AT91_REG *)     0xFFFFF214)
1424                               // (DBGU) Channel Status Register
1425 #  define AT91C_DBGU_IDR  ((AT91_REG *)     0xFFFFF20C)
1426                               // (DBGU) Interrupt Disable Register
1427 #  define AT91C_DBGU_MR   ((AT91_REG *)     0xFFFFF204)
1428                               // (DBGU) Mode Register
1429 #  define AT91C_DBGU_FNTR ((AT91_REG *)     0xFFFFF248)
1430                               // (DBGU) Force NTRST Register
1431 #  define AT91C_DBGU_C1R  ((AT91_REG *)     0xFFFFF240)
1432                               // (DBGU) Chip ID1 Register
1433 #  define AT91C_DBGU_BRGR ((AT91_REG *)     0xFFFFF220)
1434                               // (DBGU) Baud Rate Generator Register
1435 #  define AT91C_DBGU_RHR  ((AT91_REG *)     0xFFFFF218)
1436                               // (DBGU) Receiver Holding Register
1437 #  define AT91C_DBGU_IMR  ((AT91_REG *)     0xFFFFF210)
```

```
1438                                 // (DBGU) Interrupt Mask Register
1439 #  define AT91C_DBGU_IER  ((AT91_REG *)     0xFFFFF208)
1440                                 // (DBGU) Interrupt Enable Register
1441 #  define AT91C_DBGU_CR   ((AT91_REG *)     0xFFFFF200)
1442                                 // (DBGU) Control Register
1443 // ========== Register definition for PDC_DBGU peripheral ==========
1444 #  define AT91C_DBGU_TNCR ((AT91_REG *)     0xFFFFF31C)
1445                                 // (PDC_DBGU) Transmit Next Counter Register
1446 #  define AT91C_DBGU_RNCR ((AT91_REG *)     0xFFFFF314)
1447                                 // (PDC_DBGU) Receive Next Counter Register
1448 #  define AT91C_DBGU_PTCR ((AT91_REG *)     0xFFFFF320)
1449                                 // (PDC_DBGU) PDC Transfer Control Register
1450 #  define AT91C_DBGU_PTSR ((AT91_REG *)     0xFFFFF324)
1451                                 // (PDC_DBGU) PDC Transfer Status Register
1452 #  define AT91C_DBGU_RCR  ((AT91_REG *)     0xFFFFF304)
1453                                 // (PDC_DBGU) Receive Counter Register
1454 #  define AT91C_DBGU_TCR  ((AT91_REG *)     0xFFFFF30C)
1455                                 // (PDC_DBGU) Transmit Counter Register
1456 #  define AT91C_DBGU_RPR  ((AT91_REG *)     0xFFFFF300)
1457                                 // (PDC_DBGU) Receive Pointer Register
1458 #  define AT91C_DBGU_TPR  ((AT91_REG *)     0xFFFFF308)
1459                                 // (PDC_DBGU) Transmit Pointer Register
1460 #  define AT91C_DBGU_RNPR ((AT91_REG *)     0xFFFFF310)
1461                                 // (PDC_DBGU) Receive Next Pointer Register
1462 #  define AT91C_DBGU_TNPR ((AT91_REG *)     0xFFFFF318)
1463                                 // (PDC_DBGU) Transmit Next Pointer Register
1464 // ========== Register definition for PIOA peripheral ==========
1465 #  define AT91C_PIOA_IMR  ((AT91_REG *)     0xFFFFF448)
1466                                 // (PIOA) Interrupt Mask Register
1467 #  define AT91C_PIOA_IER  ((AT91_REG *)     0xFFFFF440)
1468                                 // (PIOA) Interrupt Enable Register
1469 #  define AT91C_PIOA_OWDR ((AT91_REG *)     0xFFFFF4A4)
1470                                 // (PIOA) Output Write Disable Register
1471 #  define AT91C_PIOA_ISR  ((AT91_REG *)     0xFFFFF44C)
1472                                 // (PIOA) Interrupt Status Register
1473 #  define AT91C_PIOA_PPUDR ((AT91_REG *)    0xFFFFF460)
1474                                 // (PIOA) Pull-up Disable Register
1475 #  define AT91C_PIOA_MDSR ((AT91_REG *)     0xFFFFF458)
1476                                 // (PIOA) Multi-driver Status Register
1477 #  define AT91C_PIOA_MDER ((AT91_REG *)     0xFFFFF450)
```

```
1478                                 // (PIOA) Multi-driver Enable Register
1479 #  define AT91C_PIOA_PER  ((AT91_REG *)     0xFFFFF400)
1480                                 // (PIOA) PIO Enable Register
1481 #  define AT91C_PIOA_PSR  ((AT91_REG *)     0xFFFFF408)
1482                                 // (PIOA) PIO Status Register
1483 #  define AT91C_PIOA_OER  ((AT91_REG *)     0xFFFFF410)
1484                                 // (PIOA) Output Enable Register
1485 #  define AT91C_PIOA_BSR  ((AT91_REG *)     0xFFFFF474)
1486                                 // (PIOA) Select B Register
1487 #  define AT91C_PIOA_PPUER ((AT91_REG *)    0xFFFFF464)
1488                                 // (PIOA) Pull-up Enable Register
1489 #  define AT91C_PIOA_MDDR ((AT91_REG *)     0xFFFFF454)
1490                                 // (PIOA) Multi-driver Disable Register
1491 #  define AT91C_PIOA_PDR  ((AT91_REG *)     0xFFFFF404)
1492                                 // (PIOA) PIO Disable Register
1493 #  define AT91C_PIOA_ODR  ((AT91_REG *)     0xFFFFF414)
1494                                 // (PIOA) Output Disable Register
1495 #  define AT91C_PIOA_IFDR ((AT91_REG *)     0xFFFFF424)
1496                                 // (PIOA) Input Filter Disable Register
1497 #  define AT91C_PIOA_ABSR ((AT91_REG *)     0xFFFFF478)
1498                                 // (PIOA) AB Select Status Register
1499 #  define AT91C_PIOA_ASR  ((AT91_REG *)     0xFFFFF470)
1500                                 // (PIOA) Select A Register
1501 #  define AT91C_PIOA_PPUSR ((AT91_REG *)    0xFFFFF468)
1502                                 // (PIOA) Pad Pull-up Status Register
1503 #  define AT91C_PIOA_ODSR ((AT91_REG *)     0xFFFFF438)
1504                                 // (PIOA) Output Data Status Register
1505 #  define AT91C_PIOA_SODR ((AT91_REG *)     0xFFFFF430)
1506                                 // (PIOA) Set Output Data Register
1507 #  define AT91C_PIOA_IFSR ((AT91_REG *)     0xFFFFF428)
1508                                 // (PIOA) Input Filter Status Register
1509 #  define AT91C_PIOA_IFER ((AT91_REG *)     0xFFFFF420)
1510                                 // (PIOA) Input Filter Enable Register
1511 #  define AT91C_PIOA_OSR  ((AT91_REG *)     0xFFFFF418)
1512                                 // (PIOA) Output Status Register
1513 #  define AT91C_PIOA_IDR  ((AT91_REG *)     0xFFFFF444)
1514                                 // (PIOA) Interrupt Disable Register
1515 #  define AT91C_PIOA_PDSR ((AT91_REG *)     0xFFFFF43C)
1516                                 // (PIOA) Pin Data Status Register
1517 #  define AT91C_PIOA_CODR ((AT91_REG *)     0xFFFFF434)
```

```
1518                                // (PIOA) Clear Output Data Register
1519 #  define AT91C_PIOA_OWSR ((AT91_REG *)     0xFFFFF4A8)
1520                                // (PIOA) Output Write Status Register
1521 #  define AT91C_PIOA_OWER ((AT91_REG *)     0xFFFFF4A0)
1522                                // (PIOA) Output Write Enable Register
1523 // ========== Register definition for CKGR peripheral ==========
1524 #  define AT91C_CKGR_PLLR ((AT91_REG *)     0xFFFFFC2C)
1525                                // (CKGR) PLL Register
1526 #  define AT91C_CKGR_MCFR ((AT91_REG *)     0xFFFFFC24)
1527                                // (CKGR) Main Clock  Frequency Register
1528 #  define AT91C_CKGR_MOR  ((AT91_REG *)     0xFFFFFC20)
1529                                // (CKGR) Main Oscillator Register
1530 // ========== Register definition for PMC peripheral ==========
1531 #  define AT91C_PMC_SCSR  ((AT91_REG *)     0xFFFFFC08)
1532                                // (PMC) System Clock Status Register
1533 #  define AT91C_PMC_SCER  ((AT91_REG *)     0xFFFFFC00)
1534                                // (PMC) System Clock Enable Register
1535 #  define AT91C_PMC_IMR   ((AT91_REG *)     0xFFFFFC6C)
1536                                // (PMC) Interrupt Mask Register
1537 #  define AT91C_PMC_IDR   ((AT91_REG *)     0xFFFFFC64)
1538                                // (PMC) Interrupt Disable Register
1539 #  define AT91C_PMC_PCDR  ((AT91_REG *)     0xFFFFFC14)
1540                                // (PMC) Peripheral Clock Disable Register
1541 #  define AT91C_PMC_SCDR  ((AT91_REG *)     0xFFFFFC04)
1542                                // (PMC) System Clock Disable Register
1543 #  define AT91C_PMC_SR    ((AT91_REG *)     0xFFFFFC68)
1544                                // (PMC) Status Register
1545 #  define AT91C_PMC_IER   ((AT91_REG *)     0xFFFFFC60)
1546                                // (PMC) Interrupt Enable Register
1547 #  define AT91C_PMC_MCKR  ((AT91_REG *)     0xFFFFFC30)
1548                                // (PMC) Master Clock Register
1549 #  define AT91C_PMC_MOR   ((AT91_REG *)     0xFFFFFC20)
1550                                // (PMC) Main Oscillator Register
1551 #  define AT91C_PMC_PCER  ((AT91_REG *)     0xFFFFFC10)
1552                                // (PMC) Peripheral Clock Enable Register
1553 #  define AT91C_PMC_PCSR  ((AT91_REG *)     0xFFFFFC18)
1554                                // (PMC) Peripheral Clock Status Register
1555 #  define AT91C_PMC_PLLR  ((AT91_REG *)     0xFFFFFC2C)
1556                                // (PMC) PLL Register
1557 #  define AT91C_PMC_MCFR  ((AT91_REG *)     0xFFFFFC24)
```

```
1558                                // (PMC) Main Clock  Frequency Register
1559 #  define AT91C_PMC_PCKR  ((AT91_REG *)     0xFFFFFC40)
1560                                // (PMC) Programmable Clock Register
1561 // ========== Register definition for RSTC peripheral ==========
1562 #  define AT91C_RSTC_RSR  ((AT91_REG *)     0xFFFFFD04)
1563                                // (RSTC) Reset Status Register
1564 #  define AT91C_RSTC_RMR  ((AT91_REG *)     0xFFFFFD08)
1565                                // (RSTC) Reset Mode Register
1566 #  define AT91C_RSTC_RCR  ((AT91_REG *)     0xFFFFFD00)
1567                                // (RSTC) Reset Control Register
1568 // ========== Register definition for RTTC peripheral ==========
1569 #  define AT91C_RTTC_RTSR ((AT91_REG *)     0xFFFFFD2C)
1570                                // (RTTC) Real-time Status Register
1571 #  define AT91C_RTTC_RTAR ((AT91_REG *)     0xFFFFFD24)
1572                                // (RTTC) Real-time Alarm Register
1573 #  define AT91C_RTTC_RTVR ((AT91_REG *)     0xFFFFFD28)
1574                                // (RTTC) Real-time Value Register
1575 #  define AT91C_RTTC_RTMR ((AT91_REG *)     0xFFFFFD20)
1576                                // (RTTC) Real-time Mode Register
1577 // ========== Register definition for PITC peripheral ==========
1578 #  define AT91C_PITC_PIIR ((AT91_REG *)     0xFFFFFD3C)
1579                                // (PITC) Period Interval Image Register
1580 #  define AT91C_PITC_PISR ((AT91_REG *)     0xFFFFFD34)
1581                                // (PITC) Period Interval Status Register
1582 #  define AT91C_PITC_PIVR ((AT91_REG *)     0xFFFFFD38)
1583                                // (PITC) Period Interval Value Register
1584 #  define AT91C_PITC_PIMR ((AT91_REG *)     0xFFFFFD30)
1585                                // (PITC) Period Interval Mode Register
1586 // ========== Register definition for WDTC peripheral ==========
1587 #  define AT91C_WDTC_WDMR ((AT91_REG *)     0xFFFFFD44)
1588                                // (WDTC) Watchdog Mode Register
1589 #  define AT91C_WDTC_WDSR ((AT91_REG *)     0xFFFFFD48)
1590                                // (WDTC) Watchdog Status Register
1591 #  define AT91C_WDTC_WDCR ((AT91_REG *)     0xFFFFFD40)
1592                                // (WDTC) Watchdog Control Register
1593 // ========== Register definition for MC peripheral ==========
1594 #  define AT91C_MC_FCR    ((AT91_REG *)     0xFFFFFF64)
1595                                // (MC) MC Flash Command Register
1596 #  define AT91C_MC_ASR    ((AT91_REG *)     0xFFFFFF04)
1597                                // (MC) MC Abort Status Register
```

```
1598 #  define AT91C_MC_FSR    ((AT91_REG *)     0xFFFFFF68)
1599                             // (MC) MC Flash Status Register
1600 #  define AT91C_MC_FMR    ((AT91_REG *)     0xFFFFFF60)
1601                             // (MC) MC Flash Mode Register
1602 #  define AT91C_MC_AASR   ((AT91_REG *)     0xFFFFFF08)
1603                             // (MC) MC Abort Address Status Register
1604 #  define AT91C_MC_RCR    ((AT91_REG *)     0xFFFFFF00)
1605                             // (MC) MC Remap Control Register
1606 // ========== Register definition for PDC_SPI peripheral ==========
1607 #  define AT91C_SPI_PTCR  ((AT91_REG *)     0xFFFE0120)
1608                             // (PDC_SPI) PDC Transfer Control Register
1609 #  define AT91C_SPI_TNPR  ((AT91_REG *)     0xFFFE0118)
1610                             // (PDC_SPI) Transmit Next Pointer Register
1611 #  define AT91C_SPI_RNPR  ((AT91_REG *)     0xFFFE0110)
1612                             // (PDC_SPI) Receive Next Pointer Register
1613 #  define AT91C_SPI_TPR   ((AT91_REG *)     0xFFFE0108)
1614                             // (PDC_SPI) Transmit Pointer Register
1615 #  define AT91C_SPI_RPR   ((AT91_REG *)     0xFFFE0100)
1616                             // (PDC_SPI) Receive Pointer Register
1617 #  define AT91C_SPI_PTSR  ((AT91_REG *)     0xFFFE0124)
1618                             // (PDC_SPI) PDC Transfer Status Register
1619 #  define AT91C_SPI_TNCR  ((AT91_REG *)     0xFFFE011C)
1620                             // (PDC_SPI) Transmit Next Counter Register
1621 #  define AT91C_SPI_RNCR  ((AT91_REG *)     0xFFFE0114)
1622                             // (PDC_SPI) Receive Next Counter Register
1623 #  define AT91C_SPI_TCR   ((AT91_REG *)     0xFFFE010C)
1624                             // (PDC_SPI) Transmit Counter Register
1625 #  define AT91C_SPI_RCR   ((AT91_REG *)     0xFFFE0104)
1626                             // (PDC_SPI) Receive Counter Register
1627 // ========== Register definition for SPI peripheral ==========
1628 #  define AT91C_SPI_CSR   ((AT91_REG *)     0xFFFE0030)
1629                             // (SPI) Chip Select Register
1630 #  define AT91C_SPI_IDR   ((AT91_REG *)     0xFFFE0018)
1631                             // (SPI) Interrupt Disable Register
1632 #  define AT91C_SPI_SR    ((AT91_REG *)     0xFFFE0010)
1633                             // (SPI) Status Register
1634 #  define AT91C_SPI_RDR   ((AT91_REG *)     0xFFFE0008)
1635                             // (SPI) Receive Data Register
1636 #  define AT91C_SPI_CR    ((AT91_REG *)     0xFFFE0000)
1637                             // (SPI) Control Register
```

```
1638 #  define AT91C_SPI_IMR   ((AT91_REG *)     0xFFFE001C)
1639                               // (SPI) Interrupt Mask Register
1640 #  define AT91C_SPI_IER   ((AT91_REG *)     0xFFFE0014)
1641                               // (SPI) Interrupt Enable Register
1642 #  define AT91C_SPI_TDR   ((AT91_REG *)     0xFFFE000C)
1643                               // (SPI) Transmit Data Register
1644 #  define AT91C_SPI_MR    ((AT91_REG *)     0xFFFE0004)
1645                               // (SPI) Mode Register
1646 // ========== Register definition for PDC_ADC peripheral ==========
1647 #  define AT91C_ADC_PTCR  ((AT91_REG *)     0xFFFD8120)
1648                               // (PDC_ADC) PDC Transfer Control Register
1649 #  define AT91C_ADC_TNPR  ((AT91_REG *)     0xFFFD8118)
1650                               // (PDC_ADC) Transmit Next Pointer Register
1651 #  define AT91C_ADC_RNPR  ((AT91_REG *)     0xFFFD8110)
1652                               // (PDC_ADC) Receive Next Pointer Register
1653 #  define AT91C_ADC_TPR   ((AT91_REG *)     0xFFFD8108)
1654                               // (PDC_ADC) Transmit Pointer Register
1655 #  define AT91C_ADC_RPR   ((AT91_REG *)     0xFFFD8100)
1656                               // (PDC_ADC) Receive Pointer Register
1657 #  define AT91C_ADC_PTSR  ((AT91_REG *)     0xFFFD8124)
1658                               // (PDC_ADC) PDC Transfer Status Register
1659 #  define AT91C_ADC_TNCR  ((AT91_REG *)     0xFFFD811C)
1660                               // (PDC_ADC) Transmit Next Counter Register
1661 #  define AT91C_ADC_RNCR  ((AT91_REG *)     0xFFFD8114)
1662                               // (PDC_ADC) Receive Next Counter Register
1663 #  define AT91C_ADC_TCR   ((AT91_REG *)     0xFFFD810C)
1664                               // (PDC_ADC) Transmit Counter Register
1665 #  define AT91C_ADC_RCR   ((AT91_REG *)     0xFFFD8104)
1666                               // (PDC_ADC) Receive Counter Register
1667 // ========== Register definition for ADC peripheral ==========
1668 #  define AT91C_ADC_IMR   ((AT91_REG *)     0xFFFD802C)
1669                               // (ADC) ADC Interrupt Mask Register
1670 #  define AT91C_ADC_CDR4  ((AT91_REG *)     0xFFFD8040)
1671                               // (ADC) ADC Channel Data Register 4
1672 #  define AT91C_ADC_CDR2  ((AT91_REG *)     0xFFFD8038)
1673                               // (ADC) ADC Channel Data Register 2
1674 #  define AT91C_ADC_CDR0  ((AT91_REG *)     0xFFFD8030)
1675                               // (ADC) ADC Channel Data Register 0
1676 #  define AT91C_ADC_CDR7  ((AT91_REG *)     0xFFFD804C)
1677                               // (ADC) ADC Channel Data Register 7
```

```
1678 #  define AT91C_ADC_CDR1  ((AT91_REG *)     0xFFFD8034)
1679                                 // (ADC) ADC Channel Data Register 1
1680 #  define AT91C_ADC_CDR3  ((AT91_REG *)     0xFFFD803C)
1681                                 // (ADC) ADC Channel Data Register 3
1682 #  define AT91C_ADC_CDR5  ((AT91_REG *)     0xFFFD8044)
1683                                 // (ADC) ADC Channel Data Register 5
1684 #  define AT91C_ADC_MR    ((AT91_REG *)     0xFFFD8004)
1685                                 // (ADC) ADC Mode Register
1686 #  define AT91C_ADC_CDR6  ((AT91_REG *)     0xFFFD8048)
1687                                 // (ADC) ADC Channel Data Register 6
1688 #  define AT91C_ADC_CR    ((AT91_REG *)     0xFFFD8000)
1689                                 // (ADC) ADC Control Register
1690 #  define AT91C_ADC_CHER  ((AT91_REG *)     0xFFFD8010)
1691                                 // (ADC) ADC Channel Enable Register
1692 #  define AT91C_ADC_CHSR  ((AT91_REG *)     0xFFFD8018)
1693                                 // (ADC) ADC Channel Status Register
1694 #  define AT91C_ADC_IER   ((AT91_REG *)     0xFFFD8024)
1695                                 // (ADC) ADC Interrupt Enable Register
1696 #  define AT91C_ADC_SR    ((AT91_REG *)     0xFFFD801C)
1697                                 // (ADC) ADC Status Register
1698 #  define AT91C_ADC_CHDR  ((AT91_REG *)     0xFFFD8014)
1699                                 // (ADC) ADC Channel Disable Register
1700 #  define AT91C_ADC_IDR   ((AT91_REG *)     0xFFFD8028)
1701                                 // (ADC) ADC Interrupt Disable Register
1702 #  define AT91C_ADC_LCDR  ((AT91_REG *)     0xFFFD8020)
1703                                 // (ADC) ADC Last Converted Data Register
1704 // ========== Register definition for PDC_SSC peripheral ==========
1705 #  define AT91C_SSC_PTCR  ((AT91_REG *)     0xFFFD4120)
1706                                 // (PDC_SSC) PDC Transfer Control Register
1707 #  define AT91C_SSC_TNPR  ((AT91_REG *)     0xFFFD4118)
1708                                 // (PDC_SSC) Transmit Next Pointer Register
1709 #  define AT91C_SSC_RNPR  ((AT91_REG *)     0xFFFD4110)
1710                                 // (PDC_SSC) Receive Next Pointer Register
1711 #  define AT91C_SSC_TPR   ((AT91_REG *)     0xFFFD4108)
1712                                 // (PDC_SSC) Transmit Pointer Register
1713 #  define AT91C_SSC_RPR   ((AT91_REG *)     0xFFFD4100)
1714                                 // (PDC_SSC) Receive Pointer Register
1715 #  define AT91C_SSC_PTSR  ((AT91_REG *)     0xFFFD4124)
1716                                 // (PDC_SSC) PDC Transfer Status Register
1717 #  define AT91C_SSC_TNCR  ((AT91_REG *)     0xFFFD411C)
```

```
1718                                    // (PDC_SSC) Transmit Next Counter Register
1719 #  define AT91C_SSC_RNCR  ((AT91_REG *)     0xFFFD4114)
1720                                    // (PDC_SSC) Receive Next Counter Register
1721 #  define AT91C_SSC_TCR   ((AT91_REG *)     0xFFFD410C)
1722                                    // (PDC_SSC) Transmit Counter Register
1723 #  define AT91C_SSC_RCR   ((AT91_REG *)     0xFFFD4104)
1724                                    // (PDC_SSC) Receive Counter Register
1725 // ========== Register definition for SSC peripheral ==========
1726 #  define AT91C_SSC_RFMR  ((AT91_REG *)     0xFFFD4014)
1727                                    // (SSC) Receive Frame Mode Register
1728 #  define AT91C_SSC_CMR   ((AT91_REG *)     0xFFFD4004)
1729                                    // (SSC) Clock Mode Register
1730 #  define AT91C_SSC_IDR   ((AT91_REG *)     0xFFFD4048)
1731                                    // (SSC) Interrupt Disable Register
1732 #  define AT91C_SSC_SR    ((AT91_REG *)     0xFFFD4040)
1733                                    // (SSC) Status Register
1734 #  define AT91C_SSC_RC0R  ((AT91_REG *)     0xFFFD4038)
1735                                    // (SSC) Receive Compare 0 Register
1736 #  define AT91C_SSC_RSHR  ((AT91_REG *)     0xFFFD4030)
1737                                    // (SSC) Receive Sync Holding Register
1738 #  define AT91C_SSC_RHR   ((AT91_REG *)     0xFFFD4020)
1739                                    // (SSC) Receive Holding Register
1740 #  define AT91C_SSC_TCMR  ((AT91_REG *)     0xFFFD4018)
1741                                    // (SSC) Transmit Clock Mode Register
1742 #  define AT91C_SSC_RCMR  ((AT91_REG *)     0xFFFD4010)
1743                                    // (SSC) Receive Clock ModeRegister
1744 #  define AT91C_SSC_CR    ((AT91_REG *)     0xFFFD4000)
1745                                    // (SSC) Control Register
1746 #  define AT91C_SSC_IMR   ((AT91_REG *)     0xFFFD404C)
1747                                    // (SSC) Interrupt Mask Register
1748 #  define AT91C_SSC_IER   ((AT91_REG *)     0xFFFD4044)
1749                                    // (SSC) Interrupt Enable Register
1750 #  define AT91C_SSC_RC1R  ((AT91_REG *)     0xFFFD403C)
1751                                    // (SSC) Receive Compare 1 Register
1752 #  define AT91C_SSC_TSHR  ((AT91_REG *)     0xFFFD4034)
1753                                    // (SSC) Transmit Sync Holding Register
1754 #  define AT91C_SSC_THR   ((AT91_REG *)     0xFFFD4024)
1755                                    // (SSC) Transmit Holding Register
1756 #  define AT91C_SSC_TFMR  ((AT91_REG *)     0xFFFD401C)
1757                                    // (SSC) Transmit Frame Mode Register
```

```
1758 // ========== Register definition for PDC_US1 peripheral ==========
1759 #  define AT91C_US1_PTSR  ((AT91_REG *)     0xFFFC4124)
1760                              // (PDC_US1) PDC Transfer Status Register
1761 #  define AT91C_US1_TNCR  ((AT91_REG *)     0xFFFC411C)
1762                              // (PDC_US1) Transmit Next Counter Register
1763 #  define AT91C_US1_RNCR  ((AT91_REG *)     0xFFFC4114)
1764                              // (PDC_US1) Receive Next Counter Register
1765 #  define AT91C_US1_TCR   ((AT91_REG *)     0xFFFC410C)
1766                              // (PDC_US1) Transmit Counter Register
1767 #  define AT91C_US1_RCR   ((AT91_REG *)     0xFFFC4104)
1768                              // (PDC_US1) Receive Counter Register
1769 #  define AT91C_US1_PTCR  ((AT91_REG *)     0xFFFC4120)
1770                              // (PDC_US1) PDC Transfer Control Register
1771 #  define AT91C_US1_TNPR  ((AT91_REG *)     0xFFFC4118)
1772                              // (PDC_US1) Transmit Next Pointer Register
1773 #  define AT91C_US1_RNPR  ((AT91_REG *)     0xFFFC4110)
1774                              // (PDC_US1) Receive Next Pointer Register
1775 #  define AT91C_US1_TPR   ((AT91_REG *)     0xFFFC4108)
1776                              // (PDC_US1) Transmit Pointer Register
1777 #  define AT91C_US1_RPR   ((AT91_REG *)     0xFFFC4100)
1778                              // (PDC_US1) Receive Pointer Register
1779 // ========== Register definition for US1 peripheral ==========
1780 #  define AT91C_US1_XXR   ((AT91_REG *)     0xFFFC4048)
1781                              // (US1) XON_XOFF Register
1782 #  define AT91C_US1_RHR   ((AT91_REG *)     0xFFFC4018)
1783                              // (US1) Receiver Holding Register
1784 #  define AT91C_US1_IMR   ((AT91_REG *)     0xFFFC4010)
1785                              // (US1) Interrupt Mask Register
1786 #  define AT91C_US1_IER   ((AT91_REG *)     0xFFFC4008)
1787                              // (US1) Interrupt Enable Register
1788 #  define AT91C_US1_CR    ((AT91_REG *)     0xFFFC4000)
1789                              // (US1) Control Register
1790 #  define AT91C_US1_RTOR  ((AT91_REG *)     0xFFFC4024)
1791                              // (US1) Receiver Time-out Register
1792 #  define AT91C_US1_THR   ((AT91_REG *)     0xFFFC401C)
1793                              // (US1) Transmitter Holding Register
1794 #  define AT91C_US1_CSR   ((AT91_REG *)     0xFFFC4014)
1795                              // (US1) Channel Status Register
1796 #  define AT91C_US1_IDR   ((AT91_REG *)     0xFFFC400C)
1797                              // (US1) Interrupt Disable Register
```

```
1798 #  define AT91C_US1_FIDI  ((AT91_REG *)     0xFFFC4040)
1799                             // (US1) FI_DI_Ratio Register
1800 #  define AT91C_US1_BRGR  ((AT91_REG *)     0xFFFC4020)
1801                             // (US1) Baud Rate Generator Register
1802 #  define AT91C_US1_TTGR  ((AT91_REG *)     0xFFFC4028)
1803                             // (US1) Transmitter Time-guard Register
1804 #  define AT91C_US1_IF    ((AT91_REG *)     0xFFFC404C)
1805                             // (US1) IRDA_FILTER Register
1806 #  define AT91C_US1_NER   ((AT91_REG *)     0xFFFC4044)
1807                             // (US1) Nb Errors Register
1808 #  define AT91C_US1_MR    ((AT91_REG *)     0xFFFC4004)
1809                             // (US1) Mode Register
1810 // ========== Register definition for PDC_US0 peripheral ==========
1811 #  define AT91C_US0_PTCR  ((AT91_REG *)     0xFFFC0120)
1812                             // (PDC_US0) PDC Transfer Control Register
1813 #  define AT91C_US0_TNPR  ((AT91_REG *)     0xFFFC0118)
1814                             // (PDC_US0) Transmit Next Pointer Register
1815 #  define AT91C_US0_RNPR  ((AT91_REG *)     0xFFFC0110)
1816                             // (PDC_US0) Receive Next Pointer Register
1817 #  define AT91C_US0_TPR   ((AT91_REG *)     0xFFFC0108)
1818                             // (PDC_US0) Transmit Pointer Register
1819 #  define AT91C_US0_RPR   ((AT91_REG *)     0xFFFC0100)
1820                             // (PDC_US0) Receive Pointer Register
1821 #  define AT91C_US0_PTSR  ((AT91_REG *)     0xFFFC0124)
1822                             // (PDC_US0) PDC Transfer Status Register
1823 #  define AT91C_US0_TNCR  ((AT91_REG *)     0xFFFC011C)
1824                             // (PDC_US0) Transmit Next Counter Register
1825 #  define AT91C_US0_RNCR  ((AT91_REG *)     0xFFFC0114)
1826                             // (PDC_US0) Receive Next Counter Register
1827 #  define AT91C_US0_TCR   ((AT91_REG *)     0xFFFC010C)
1828                             // (PDC_US0) Transmit Counter Register
1829 #  define AT91C_US0_RCR   ((AT91_REG *)     0xFFFC0104)
1830                             // (PDC_US0) Receive Counter Register
1831 // ========== Register definition for US0 peripheral ==========
1832 #  define AT91C_US0_TTGR  ((AT91_REG *)     0xFFFC0028)
1833                             // (US0) Transmitter Time-guard Register
1834 #  define AT91C_US0_BRGR  ((AT91_REG *)     0xFFFC0020)
1835                             // (US0) Baud Rate Generator Register
1836 #  define AT91C_US0_RHR   ((AT91_REG *)     0xFFFC0018)
1837                             // (US0) Receiver Holding Register
```

```
1838 #  define AT91C_US0_IMR   ((AT91_REG *)     0xFFFC0010)
1839                               // (US0) Interrupt Mask Register
1840 #  define AT91C_US0_NER   ((AT91_REG *)     0xFFFC0044)
1841                               // (US0) Nb Errors Register
1842 #  define AT91C_US0_RTOR  ((AT91_REG *)     0xFFFC0024)
1843                               // (US0) Receiver Time-out Register
1844 #  define AT91C_US0_XXR   ((AT91_REG *)     0xFFFC0048)
1845                               // (US0) XON_XOFF Register
1846 #  define AT91C_US0_FIDI  ((AT91_REG *)     0xFFFC0040)
1847                               // (US0) FI_DI_Ratio Register
1848 #  define AT91C_US0_CR    ((AT91_REG *)     0xFFFC0000)
1849                               // (US0) Control Register
1850 #  define AT91C_US0_IER   ((AT91_REG *)     0xFFFC0008)
1851                               // (US0) Interrupt Enable Register
1852 #  define AT91C_US0_IF    ((AT91_REG *)     0xFFFC004C)
1853                               // (US0) IRDA_FILTER Register
1854 #  define AT91C_US0_MR    ((AT91_REG *)     0xFFFC0004)
1855                               // (US0) Mode Register
1856 #  define AT91C_US0_IDR   ((AT91_REG *)     0xFFFC000C)
1857                               // (US0) Interrupt Disable Register
1858 #  define AT91C_US0_CSR   ((AT91_REG *)     0xFFFC0014)
1859                               // (US0) Channel Status Register
1860 #  define AT91C_US0_THR   ((AT91_REG *)     0xFFFC001C)
1861                               // (US0) Transmitter Holding Register
1862 // ========== Register definition for TWI peripheral ==========
1863 #  define AT91C_TWI_RHR   ((AT91_REG *)     0xFFFB8030)
1864                               // (TWI) Receive Holding Register
1865 #  define AT91C_TWI_IDR   ((AT91_REG *)     0xFFFB8028)
1866                               // (TWI) Interrupt Disable Register
1867 #  define AT91C_TWI_SR    ((AT91_REG *)     0xFFFB8020)
1868                               // (TWI) Status Register
1869 #  define AT91C_TWI_CWGR  ((AT91_REG *)     0xFFFB8010)
1870                               // (TWI) Clock Waveform Generator Register
1871 #  define AT91C_TWI_SMR   ((AT91_REG *)     0xFFFB8008)
1872                               // (TWI) Slave Mode Register
1873 #  define AT91C_TWI_CR    ((AT91_REG *)     0xFFFB8000)
1874                               // (TWI) Control Register
1875 #  define AT91C_TWI_THR   ((AT91_REG *)     0xFFFB8034)
1876                               // (TWI) Transmit Holding Register
1877 #  define AT91C_TWI_IMR   ((AT91_REG *)     0xFFFB802C)
```

```
1878                                // (TWI) Interrupt Mask Register
1879 #  define AT91C_TWI_IER    ((AT91_REG *)     0xFFFB8024)
1880                                // (TWI) Interrupt Enable Register
1881 #  define AT91C_TWI_IADR   ((AT91_REG *)     0xFFFB800C)
1882                                // (TWI) Internal Address Register
1883 #  define AT91C_TWI_MMR    ((AT91_REG *)     0xFFFB8004)
1884                                // (TWI) Master Mode Register
1885 // ========== Register definition for TC2 peripheral ==========
1886 #  define AT91C_TC2_IMR    ((AT91_REG *)     0xFFFA00AC)
1887                                // (TC2) Interrupt Mask Register
1888 #  define AT91C_TC2_IER    ((AT91_REG *)     0xFFFA00A4)
1889                                // (TC2) Interrupt Enable Register
1890 #  define AT91C_TC2_RC     ((AT91_REG *)     0xFFFA009C)
1891                                // (TC2) Register C
1892 #  define AT91C_TC2_RA     ((AT91_REG *)     0xFFFA0094)
1893                                // (TC2) Register A
1894 #  define AT91C_TC2_CMR    ((AT91_REG *)     0xFFFA0084)
1895                                // (TC2) Channel Mode Register (Capture Mode /
    …                           Waveform Mode)
1896 #  define AT91C_TC2_IDR    ((AT91_REG *)     0xFFFA00A8)
1897                                // (TC2) Interrupt Disable Register
1898 #  define AT91C_TC2_SR     ((AT91_REG *)     0xFFFA00A0)
1899                                // (TC2) Status Register
1900 #  define AT91C_TC2_RB     ((AT91_REG *)     0xFFFA0098)
1901                                // (TC2) Register B
1902 #  define AT91C_TC2_CV     ((AT91_REG *)     0xFFFA0090)
1903                                // (TC2) Counter Value
1904 #  define AT91C_TC2_CCR    ((AT91_REG *)     0xFFFA0080)
1905                                // (TC2) Channel Control Register
1906 // ========== Register definition for TC1 peripheral ==========
1907 #  define AT91C_TC1_IMR    ((AT91_REG *)     0xFFFA006C)
1908                                // (TC1) Interrupt Mask Register
1909 #  define AT91C_TC1_IER    ((AT91_REG *)     0xFFFA0064)
1910                                // (TC1) Interrupt Enable Register
1911 #  define AT91C_TC1_RC     ((AT91_REG *)     0xFFFA005C)
1912                                // (TC1) Register C
1913 #  define AT91C_TC1_RA     ((AT91_REG *)     0xFFFA0054)
1914                                // (TC1) Register A
1915 #  define AT91C_TC1_CMR    ((AT91_REG *)     0xFFFA0044)
1916                                // (TC1) Channel Mode Register (Capture Mode /
```

```
1916…                                Waveform Mode)
1917 #  define AT91C_TC1_IDR   ((AT91_REG *)     0xFFFA0068)
1918                               // (TC1) Interrupt Disable Register
1919 #  define AT91C_TC1_SR    ((AT91_REG *)     0xFFFA0060)
1920                               // (TC1) Status Register
1921 #  define AT91C_TC1_RB    ((AT91_REG *)     0xFFFA0058)
1922                               // (TC1) Register B
1923 #  define AT91C_TC1_CV    ((AT91_REG *)     0xFFFA0050)
1924                               // (TC1) Counter Value
1925 #  define AT91C_TC1_CCR   ((AT91_REG *)     0xFFFA0040)
1926                               // (TC1) Channel Control Register
1927 // ========== Register definition for TC0 peripheral ==========
1928 #  define AT91C_TC0_IMR   ((AT91_REG *)     0xFFFA002C)
1929                               // (TC0) Interrupt Mask Register
1930 #  define AT91C_TC0_IER   ((AT91_REG *)     0xFFFA0024)
1931                               // (TC0) Interrupt Enable Register
1932 #  define AT91C_TC0_RC    ((AT91_REG *)     0xFFFA001C)
1933                               // (TC0) Register C
1934 #  define AT91C_TC0_RA    ((AT91_REG *)     0xFFFA0014)
1935                               // (TC0) Register A
1936 #  define AT91C_TC0_CMR   ((AT91_REG *)     0xFFFA0004)
1937                               // (TC0) Channel Mode Register (Capture Mode /
…                                  Waveform Mode)
1938 #  define AT91C_TC0_IDR   ((AT91_REG *)     0xFFFA0028)
1939                               // (TC0) Interrupt Disable Register
1940 #  define AT91C_TC0_SR    ((AT91_REG *)     0xFFFA0020)
1941                               // (TC0) Status Register
1942 #  define AT91C_TC0_RB    ((AT91_REG *)     0xFFFA0018)
1943                               // (TC0) Register B
1944 #  define AT91C_TC0_CV    ((AT91_REG *)     0xFFFA0010)
1945                               // (TC0) Counter Value
1946 #  define AT91C_TC0_CCR   ((AT91_REG *)     0xFFFA0000)
1947                               // (TC0) Channel Control Register
1948 // ========== Register definition for TCB peripheral ==========
1949 #  define AT91C_TCB_BMR   ((AT91_REG *)     0xFFFA00C4)
1950                               // (TCB) TC Block Mode Register
1951 #  define AT91C_TCB_BCR   ((AT91_REG *)     0xFFFA00C0)
1952                               // (TCB) TC Block Control Register
1953 // ========== Register definition for PWMC_CH3 peripheral ==========
1954 #  define AT91C_CH3_CUPDR ((AT91_REG *)     0xFFFCC270)
```

```
1955                                 // (PWMC_CH3) Channel Update Register
1956 #  define AT91C_CH3_CPRDR ((AT91_REG *)     0xFFFCC268)
1957                                 // (PWMC_CH3) Channel Period Register
1958 #  define AT91C_CH3_CMR   ((AT91_REG *)     0xFFFCC260)
1959                                 // (PWMC_CH3) Channel Mode Register
1960 #  define AT91C_CH3_Reserved ((AT91_REG *)  0xFFFCC274) // (PWMC_CH3) Reserved
1961 #  define AT91C_CH3_CCNTR ((AT91_REG *)     0xFFFCC26C)
1962                                 // (PWMC_CH3) Channel Counter Register
1963 #  define AT91C_CH3_CDTYR ((AT91_REG *)     0xFFFCC264)
1964                                 // (PWMC_CH3) Channel Duty Cycle Register
1965 // ========== Register definition for PWMC_CH2 peripheral ==========
1966 #  define AT91C_CH2_CUPDR ((AT91_REG *)     0xFFFCC250)
1967                                 // (PWMC_CH2) Channel Update Register
1968 #  define AT91C_CH2_CPRDR ((AT91_REG *)     0xFFFCC248)
1969                                 // (PWMC_CH2) Channel Period Register
1970 #  define AT91C_CH2_CMR   ((AT91_REG *)     0xFFFCC240)
1971                                 // (PWMC_CH2) Channel Mode Register
1972 #  define AT91C_CH2_Reserved ((AT91_REG *)  0xFFFCC254) // (PWMC_CH2) Reserved
1973 #  define AT91C_CH2_CCNTR ((AT91_REG *)     0xFFFCC24C)
1974                                 // (PWMC_CH2) Channel Counter Register
1975 #  define AT91C_CH2_CDTYR ((AT91_REG *)     0xFFFCC244)
1976                                 // (PWMC_CH2) Channel Duty Cycle Register
1977 // ========== Register definition for PWMC_CH1 peripheral ==========
1978 #  define AT91C_CH1_CUPDR ((AT91_REG *)     0xFFFCC230)
1979                                 // (PWMC_CH1) Channel Update Register
1980 #  define AT91C_CH1_CPRDR ((AT91_REG *)     0xFFFCC228)
1981                                 // (PWMC_CH1) Channel Period Register
1982 #  define AT91C_CH1_CMR   ((AT91_REG *)     0xFFFCC220)
1983                                 // (PWMC_CH1) Channel Mode Register
1984 #  define AT91C_CH1_Reserved ((AT91_REG *)  0xFFFCC234) // (PWMC_CH1) Reserved
1985 #  define AT91C_CH1_CCNTR ((AT91_REG *)     0xFFFCC22C)
1986                                 // (PWMC_CH1) Channel Counter Register
1987 #  define AT91C_CH1_CDTYR ((AT91_REG *)     0xFFFCC224)
1988                                 // (PWMC_CH1) Channel Duty Cycle Register
1989 // ========== Register definition for PWMC_CH0 peripheral ==========
1990 #  define AT91C_CH0_CUPDR ((AT91_REG *)     0xFFFCC210)
1991                                 // (PWMC_CH0) Channel Update Register
1992 #  define AT91C_CH0_CPRDR ((AT91_REG *)     0xFFFCC208)
1993                                 // (PWMC_CH0) Channel Period Register
1994 #  define AT91C_CH0_CMR   ((AT91_REG *)     0xFFFCC200)
```

```
1995                                // (PWMC_CH0) Channel Mode Register
1996 #  define AT91C_CH0_Reserved ((AT91_REG *)  0xFFFCC214) // (PWMC_CH0) Reserved
1997 #  define AT91C_CH0_CCNTR ((AT91_REG *)     0xFFFCC20C)
1998                                // (PWMC_CH0) Channel Counter Register
1999 #  define AT91C_CH0_CDTYR ((AT91_REG *)     0xFFFCC204)
2000                                // (PWMC_CH0) Channel Duty Cycle Register
2001 // ========== Register definition for PWMC peripheral ==========
2002 #  define AT91C_PWMC_VR   ((AT91_REG *)     0xFFFCC0FC)
2003                                // (PWMC) PWMC Version Register
2004 #  define AT91C_PWMC_ISR  ((AT91_REG *)     0xFFFCC01C)
2005                                // (PWMC) PWMC Interrupt Status Register
2006 #  define AT91C_PWMC_IDR  ((AT91_REG *)     0xFFFCC014)
2007                                // (PWMC) PWMC Interrupt Disable Register
2008 #  define AT91C_PWMC_SR   ((AT91_REG *)     0xFFFCC00C)
2009                                // (PWMC) PWMC Status Register
2010 #  define AT91C_PWMC_ENA  ((AT91_REG *)     0xFFFCC004)
2011                                // (PWMC) PWMC Enable Register
2012 #  define AT91C_PWMC_IMR  ((AT91_REG *)     0xFFFCC018)
2013                                // (PWMC) PWMC Interrupt Mask Register
2014 #  define AT91C_PWMC_MR   ((AT91_REG *)     0xFFFCC000)
2015                                // (PWMC) PWMC Mode Register
2016 #  define AT91C_PWMC_DIS  ((AT91_REG *)     0xFFFCC008)
2017                                // (PWMC) PWMC Disable Register
2018 #  define AT91C_PWMC_IER  ((AT91_REG *)     0xFFFCC010)
2019                                // (PWMC) PWMC Interrupt Enable Register
2020 // ========== Register definition for UDP peripheral ==========
2021 #  define AT91C_UDP_ISR   ((AT91_REG *)     0xFFFB001C)
2022                                // (UDP) Interrupt Status Register
2023 #  define AT91C_UDP_IDR   ((AT91_REG *)     0xFFFB0014)
2024                                // (UDP) Interrupt Disable Register
2025 #  define AT91C_UDP_GLBSTATE ((AT91_REG *)  0xFFFB0004) // (UDP) Global State
   … Register
2026 #  define AT91C_UDP_FDR   ((AT91_REG *)     0xFFFB0050)
2027                                // (UDP) Endpoint FIFO Data Register
2028 #  define AT91C_UDP_CSR   ((AT91_REG *)     0xFFFB0030)
2029                                // (UDP) Endpoint Control and Status Register
2030 #  define AT91C_UDP_RSTEP ((AT91_REG *)     0xFFFB0028)
2031                                // (UDP) Reset Endpoint Register
2032 #  define AT91C_UDP_ICR   ((AT91_REG *)     0xFFFB0020)
2033                                // (UDP) Interrupt Clear Register
```

```
2034 #  define AT91C_UDP_IMR   ((AT91_REG *)     0xFFFB0018)
2035                              // (UDP) Interrupt Mask Register
2036 #  define AT91C_UDP_IER   ((AT91_REG *)     0xFFFB0010)
2037                              // (UDP) Interrupt Enable Register
2038 #  define AT91C_UDP_FADDR ((AT91_REG *)     0xFFFB0008)
2039                              // (UDP) Function Address Register
2040 #  define AT91C_UDP_NUM   ((AT91_REG *)     0xFFFB0000)
2041                              // (UDP) Frame Number Register
2042
2043 //
… ********************************************************************************
2044 //               PIO DEFINITIONS FOR AT91SAM7S64
2045 //
… ********************************************************************************
2046 #  define AT91C_PIO_PA0        ((unsigned int) 1 <<  0) // Pin Controlled by
… PA0
2047 #  define AT91C_PA0_PWM0     ((unsigned int) AT91C_PIO_PA0) //  PWM Channel 0
2048 #  define AT91C_PA0_TIOA0    ((unsigned int) AT91C_PIO_PA0) //  Timer Counter 0
… Multipurpose Timer I/O Pin A
2049 #  define AT91C_PIO_PA1        ((unsigned int) 1 <<  1) // Pin Controlled by
… PA1
2050 #  define AT91C_PA1_PWM1     ((unsigned int) AT91C_PIO_PA1) //  PWM Channel 1
2051 #  define AT91C_PA1_TIOB0    ((unsigned int) AT91C_PIO_PA1) //  Timer Counter 0
… Multipurpose Timer I/O Pin B
2052 #  define AT91C_PIO_PA10       ((unsigned int) 1 << 10) // Pin Controlled by
… PA10
2053 #  define AT91C_PA10_DTXD     ((unsigned int) AT91C_PIO_PA10)   //  DBGU Debug
… Transmit Data
2054 #  define AT91C_PA10_NPCS2    ((unsigned int) AT91C_PIO_PA10)   //  SPI
… Peripheral Chip Select 2
2055 #  define AT91C_PIO_PA11       ((unsigned int) 1 << 11) // Pin Controlled by
… PA11
2056 #  define AT91C_PA11_NPCS0    ((unsigned int) AT91C_PIO_PA11)   //  SPI
… Peripheral Chip Select 0
2057 #  define AT91C_PA11_PWM0     ((unsigned int) AT91C_PIO_PA11)   //  PWM Channel
… 0
2058 #  define AT91C_PIO_PA12       ((unsigned int) 1 << 12) // Pin Controlled by
… PA12
2059 #  define AT91C_PA12_MISO     ((unsigned int) AT91C_PIO_PA12)   //  SPI Master
… In Slave
```

```
2060 #  define AT91C_PA12_PWM1     ((unsigned int) AT91C_PIO_PA12)   //  PWM Channel
   … 1
2061 #  define AT91C_PIO_PA13       ((unsigned int) 1 << 13) // Pin Controlled by
   … PA13
2062 #  define AT91C_PA13_MOSI     ((unsigned int) AT91C_PIO_PA13)   //  SPI Master
   … Out Slave
2063 #  define AT91C_PA13_PWM2     ((unsigned int) AT91C_PIO_PA13)   //  PWM Channel
   … 2
2064 #  define AT91C_PIO_PA14       ((unsigned int) 1 << 14) // Pin Controlled by
   … PA14
2065 #  define AT91C_PA14_SPCK     ((unsigned int) AT91C_PIO_PA14)   //  SPI Serial
   … Clock
2066 #  define AT91C_PA14_PWM3     ((unsigned int) AT91C_PIO_PA14)   //  PWM Channel
   … 3
2067 #  define AT91C_PIO_PA15       ((unsigned int) 1 << 15) // Pin Controlled by
   … PA15
2068 #  define AT91C_PA15_TF       ((unsigned int) AT91C_PIO_PA15)   //  SSC
   … Transmit Frame Sync
2069 #  define AT91C_PA15_TIOA1    ((unsigned int) AT91C_PIO_PA15)   //  Timer
   … Counter 1 Multipurpose Timer I/O Pin A
2070 #  define AT91C_PIO_PA16       ((unsigned int) 1 << 16) // Pin Controlled by
   … PA16
2071 #  define AT91C_PA16_TK       ((unsigned int) AT91C_PIO_PA16)   //  SSC
   … Transmit Clock
2072 #  define AT91C_PA16_TIOB1    ((unsigned int) AT91C_PIO_PA16)   //  Timer
   … Counter 1 Multipurpose Timer I/O Pin B
2073 #  define AT91C_PIO_PA17       ((unsigned int) 1 << 17) // Pin Controlled by
   … PA17
2074 #  define AT91C_PA17_TD       ((unsigned int) AT91C_PIO_PA17)   //  SSC
   … Transmit data
2075 #  define AT91C_PA17_PCK1     ((unsigned int) AT91C_PIO_PA17)   //  PMC
   … Programmable Clock Output 1
2076 #  define AT91C_PIO_PA18       ((unsigned int) 1 << 18) // Pin Controlled by
   … PA18
2077 #  define AT91C_PA18_RD       ((unsigned int) AT91C_PIO_PA18)   //  SSC Receive
   … Data
2078 #  define AT91C_PA18_PCK2     ((unsigned int) AT91C_PIO_PA18)   //  PMC
   … Programmable Clock Output 2
2079 #  define AT91C_PIO_PA19       ((unsigned int) 1 << 19) // Pin Controlled by
   … PA19
```

```
2080 #  define AT91C_PA19_RK       ((unsigned int) AT91C_PIO_PA19)   //  SSC Receive
   … Clock
2081 #  define AT91C_PA19_FIQ      ((unsigned int) AT91C_PIO_PA19)   //  AIC Fast
   … Interrupt Input
2082 #  define AT91C_PIO_PA2        ((unsigned int) 1 <<  2) // Pin Controlled by
   … PA2
2083 #  define AT91C_PA2_PWM2     ((unsigned int) AT91C_PIO_PA2) //  PWM Channel 2
2084 #  define AT91C_PA2_SCK0     ((unsigned int) AT91C_PIO_PA2) //  USART 0 Serial
   … Clock
2085 #  define AT91C_PIO_PA20       ((unsigned int) 1 << 20) // Pin Controlled by
   … PA20
2086 #  define AT91C_PA20_RF       ((unsigned int) AT91C_PIO_PA20)   //  SSC Receive
   … Frame Sync
2087 #  define AT91C_PA20_IRQ0     ((unsigned int) AT91C_PIO_PA20)   //  External
   … Interrupt 0
2088 #  define AT91C_PIO_PA21       ((unsigned int) 1 << 21) // Pin Controlled by
   … PA21
2089 #  define AT91C_PA21_RXD1     ((unsigned int) AT91C_PIO_PA21)   //  USART 1
   … Receive Data
2090 #  define AT91C_PA21_PCK1     ((unsigned int) AT91C_PIO_PA21)   //  PMC
   … Programmable Clock Output 1
2091 #  define AT91C_PIO_PA22       ((unsigned int) 1 << 22) // Pin Controlled by
   … PA22
2092 #  define AT91C_PA22_TXD1     ((unsigned int) AT91C_PIO_PA22)   //  USART 1
   … Transmit Data
2093 #  define AT91C_PA22_NPCS3    ((unsigned int) AT91C_PIO_PA22)   //  SPI
   … Peripheral Chip Select 3
2094 #  define AT91C_PIO_PA23       ((unsigned int) 1 << 23) // Pin Controlled by
   … PA23
2095 #  define AT91C_PA23_SCK1     ((unsigned int) AT91C_PIO_PA23)   //  USART 1
   … Serial Clock
2096 #  define AT91C_PA23_PWM0     ((unsigned int) AT91C_PIO_PA23)   //  PWM Channel
   … 0
2097 #  define AT91C_PIO_PA24       ((unsigned int) 1 << 24) // Pin Controlled by
   … PA24
2098 #  define AT91C_PA24_RTS1     ((unsigned int) AT91C_PIO_PA24)   //  USART 1
   … Ready To Send
2099 #  define AT91C_PA24_PWM1     ((unsigned int) AT91C_PIO_PA24)   //  PWM Channel
   … 1
2100 #  define AT91C_PIO_PA25       ((unsigned int) 1 << 25) // Pin Controlled by
```

```
2100… PA25
2101 #  define AT91C_PA25_CTS1     ((unsigned int) AT91C_PIO_PA25)   //  USART 1
   … Clear To Send
2102 #  define AT91C_PA25_PWM2     ((unsigned int) AT91C_PIO_PA25)   //  PWM Channel
   … 2
2103 #  define AT91C_PIO_PA26       ((unsigned int) 1 << 26) // Pin Controlled by
   … PA26
2104 #  define AT91C_PA26_DCD1     ((unsigned int) AT91C_PIO_PA26)   //  USART 1
   … Data Carrier Detect
2105 #  define AT91C_PA26_TIOA2    ((unsigned int) AT91C_PIO_PA26)   //  Timer
   … Counter 2 Multipurpose Timer I/O Pin A
2106 #  define AT91C_PIO_PA27       ((unsigned int) 1 << 27) // Pin Controlled by
   … PA27
2107 #  define AT91C_PA27_DTR1     ((unsigned int) AT91C_PIO_PA27)   //  USART 1
   … Data Terminal ready
2108 #  define AT91C_PA27_TIOB2    ((unsigned int) AT91C_PIO_PA27)   //  Timer
   … Counter 2 Multipurpose Timer I/O Pin B
2109 #  define AT91C_PIO_PA28       ((unsigned int) 1 << 28) // Pin Controlled by
   … PA28
2110 #  define AT91C_PA28_DSR1     ((unsigned int) AT91C_PIO_PA28)   //  USART 1
   … Data Set ready
2111 #  define AT91C_PA28_TCLK1    ((unsigned int) AT91C_PIO_PA28)   //  Timer
   … Counter 1 external clock input
2112 #  define AT91C_PIO_PA29       ((unsigned int) 1 << 29) // Pin Controlled by
   … PA29
2113 #  define AT91C_PA29_RI1      ((unsigned int) AT91C_PIO_PA29)   //  USART 1
   … Ring Indicator
2114 #  define AT91C_PA29_TCLK2    ((unsigned int) AT91C_PIO_PA29)   //  Timer
   … Counter 2 external clock input
2115 #  define AT91C_PIO_PA3        ((unsigned int) 1 <<  3) // Pin Controlled by
   … PA3
2116 #  define AT91C_PA3_TWD      ((unsigned int) AT91C_PIO_PA3) //  TWI Two-wire
   … Serial Data
2117 #  define AT91C_PA3_NPCS3    ((unsigned int) AT91C_PIO_PA3) //  SPI Peripheral
   … Chip Select 3
2118 #  define AT91C_PIO_PA30       ((unsigned int) 1 << 30) // Pin Controlled by
   … PA30
2119 #  define AT91C_PA30_IRQ1     ((unsigned int) AT91C_PIO_PA30)   //  External
   … Interrupt 1
2120 #  define AT91C_PA30_NPCS2    ((unsigned int) AT91C_PIO_PA30)   //  SPI
```

```
2120… Peripheral Chip Select 2
2121 #  define AT91C_PIO_PA31       ((unsigned int) 1 << 31) // Pin Controlled by
…    PA31
2122 #  define AT91C_PA31_NPCS1    ((unsigned int) AT91C_PIO_PA31)   //  SPI
…    Peripheral Chip Select 1
2123 #  define AT91C_PA31_PCK2     ((unsigned int) AT91C_PIO_PA31)   //  PMC
…    Programmable Clock Output 2
2124 #  define AT91C_PIO_PA4        ((unsigned int) 1 <<  4) // Pin Controlled by
…    PA4
2125 #  define AT91C_PA4_TWCK     ((unsigned int) AT91C_PIO_PA4) //  TWI Two-wire
…    Serial Clock
2126 #  define AT91C_PA4_TCLK0    ((unsigned int) AT91C_PIO_PA4) //  Timer Counter 0
…    external clock input
2127 #  define AT91C_PIO_PA5        ((unsigned int) 1 <<  5) // Pin Controlled by
…    PA5
2128 #  define AT91C_PA5_RXD0     ((unsigned int) AT91C_PIO_PA5) //  USART 0 Receive
…    Data
2129 #  define AT91C_PA5_NPCS3    ((unsigned int) AT91C_PIO_PA5) //  SPI Peripheral
…    Chip Select 3
2130 #  define AT91C_PIO_PA6        ((unsigned int) 1 <<  6) // Pin Controlled by
…    PA6
2131 #  define AT91C_PA6_TXD0     ((unsigned int) AT91C_PIO_PA6) //  USART 0
…    Transmit Data
2132 #  define AT91C_PA6_PCK0     ((unsigned int) AT91C_PIO_PA6) //  PMC
…    Programmable Clock Output 0
2133 #  define AT91C_PIO_PA7        ((unsigned int) 1 <<  7) // Pin Controlled by
…    PA7
2134 #  define AT91C_PA7_RTS0     ((unsigned int) AT91C_PIO_PA7) //  USART 0 Ready
…    To Send
2135 #  define AT91C_PA7_PWM3     ((unsigned int) AT91C_PIO_PA7) //  PWM Channel 3
2136 #  define AT91C_PIO_PA8        ((unsigned int) 1 <<  8) // Pin Controlled by
…    PA8
2137 #  define AT91C_PA8_CTS0     ((unsigned int) AT91C_PIO_PA8) //  USART 0 Clear
…    To Send
2138 #  define AT91C_PA8_ADTRG    ((unsigned int) AT91C_PIO_PA8) //  ADC External
…    Trigger
2139 #  define AT91C_PIO_PA9        ((unsigned int) 1 <<  9) // Pin Controlled by
…    PA9
2140 #  define AT91C_PA9_DRXD     ((unsigned int) AT91C_PIO_PA9) //  DBGU Debug
…    Receive Data
```

```
2141 #  define AT91C_PA9_NPCS1    ((unsigned int) AT91C_PIO_PA9) //  SPI Peripheral
   … Chip Select 1
2142
2143 //
   … *****************************************************************************
2144 //               PERIPHERAL ID DEFINITIONS FOR AT91SAM7S64
2145 //
   … *****************************************************************************
2146 #  define AT91C_ID_FIQ    ((unsigned int)  0)   // Advanced Interrupt
   … Controller (FIQ)
2147 #  define AT91C_ID_SYS    ((unsigned int)  1)   // System Peripheral
2148 #  define AT91C_ID_PIOA   ((unsigned int)  2)   // Parallel IO Controller
2149 #  define AT91C_ID_3_Reserved ((unsigned int)  3)
2150                          // Reserved
2151 #  define AT91C_ID_ADC    ((unsigned int)  4)   // Analog-to-Digital Converter
2152 #  define AT91C_ID_SPI    ((unsigned int)  5)   // Serial Peripheral Interface
2153 #  define AT91C_ID_US0    ((unsigned int)  6)   // USART 0
2154 #  define AT91C_ID_US1    ((unsigned int)  7)   // USART 1
2155 #  define AT91C_ID_SSC    ((unsigned int)  8)   // Serial Synchronous
   … Controller
2156 #  define AT91C_ID_TWI    ((unsigned int)  9)   // Two-Wire Interface
2157 #  define AT91C_ID_PWMC   ((unsigned int) 10)   // PWM Controller
2158 #  define AT91C_ID_UDP    ((unsigned int) 11)   // USB Device Port
2159 #  define AT91C_ID_TC0    ((unsigned int) 12)   // Timer Counter 0
2160 #  define AT91C_ID_TC1    ((unsigned int) 13)   // Timer Counter 1
2161 #  define AT91C_ID_TC2    ((unsigned int) 14)   // Timer Counter 2
2162 #  define AT91C_ID_15_Reserved ((unsigned int) 15)  // Reserved
2163 #  define AT91C_ID_16_Reserved ((unsigned int) 16)  // Reserved
2164 #  define AT91C_ID_17_Reserved ((unsigned int) 17)  // Reserved
2165 #  define AT91C_ID_18_Reserved ((unsigned int) 18)  // Reserved
2166 #  define AT91C_ID_19_Reserved ((unsigned int) 19)  // Reserved
2167 #  define AT91C_ID_20_Reserved ((unsigned int) 20)  // Reserved
2168 #  define AT91C_ID_21_Reserved ((unsigned int) 21)  // Reserved
2169 #  define AT91C_ID_22_Reserved ((unsigned int) 22)  // Reserved
2170 #  define AT91C_ID_23_Reserved ((unsigned int) 23)  // Reserved
2171 #  define AT91C_ID_24_Reserved ((unsigned int) 24)  // Reserved
2172 #  define AT91C_ID_25_Reserved ((unsigned int) 25)  // Reserved
2173 #  define AT91C_ID_26_Reserved ((unsigned int) 26)  // Reserved
2174 #  define AT91C_ID_27_Reserved ((unsigned int) 27)  // Reserved
2175 #  define AT91C_ID_28_Reserved ((unsigned int) 28)  // Reserved
```

```
2176 #  define AT91C_ID_29_Reserved ((unsigned int) 29)  // Reserved
2177 #  define AT91C_ID_IRQ0   ((unsigned int) 30)   // Advanced Interrupt
   … Controller (IRQ0)
2178 #  define AT91C_ID_IRQ1   ((unsigned int) 31)   // Advanced Interrupt
   … Controller (IRQ1)
2179
2180 //
   …*****************************************************************************
2181 //               BASE ADDRESS DEFINITIONS FOR AT91SAM7S64
2182 //
   …*****************************************************************************
2183 #  define AT91C_BASE_SYSC      ((AT91PS_SYSC)   0xFFFFF000) // (SYSC) Base
   … Address
2184 #  define AT91C_BASE_AIC       ((AT91PS_AIC)    0xFFFFF000) // (AIC) Base
   … Address
2185 #  define AT91C_BASE_DBGU      ((AT91PS_DBGU)   0xFFFFF200) // (DBGU) Base
   … Address
2186 #  define AT91C_BASE_PDC_DBGU  ((AT91PS_PDC)    0xFFFFF300) // (PDC_DBGU) Base
   … Address
2187 #  define AT91C_BASE_PIOA      ((AT91PS_PIO)    0xFFFFF400) // (PIOA) Base
   … Address
2188 #  define AT91C_BASE_CKGR      ((AT91PS_CKGR)   0xFFFFFC20) // (CKGR) Base
   … Address
2189 #  define AT91C_BASE_PMC       ((AT91PS_PMC)    0xFFFFFC00) // (PMC) Base
   … Address
2190 #  define AT91C_BASE_RSTC      ((AT91PS_RSTC)   0xFFFFFD00) // (RSTC) Base
   … Address
2191 #  define AT91C_BASE_RTTC      ((AT91PS_RTTC)   0xFFFFFD20) // (RTTC) Base
   … Address
2192 #  define AT91C_BASE_PITC      ((AT91PS_PITC)   0xFFFFFD30) // (PITC) Base
   … Address
2193 #  define AT91C_BASE_WDTC      ((AT91PS_WDTC)   0xFFFFFD40) // (WDTC) Base
   … Address
2194 #  define AT91C_BASE_MC        ((AT91PS_MC)     0xFFFFFF00) // (MC) Base
   … Address
2195 #  define AT91C_BASE_PDC_SPI   ((AT91PS_PDC)    0xFFFE0100) // (PDC_SPI) Base
   … Address
2196 #  define AT91C_BASE_SPI       ((AT91PS_SPI)    0xFFFE0000) // (SPI) Base
   … Address
2197 #  define AT91C_BASE_PDC_ADC   ((AT91PS_PDC)    0xFFFD8100) // (PDC_ADC) Base
```

```
2197… Address
2198  #  define AT91C_BASE_ADC       ((AT91PS_ADC)    0xFFFD8000) // (ADC) Base
   … Address
2199  #  define AT91C_BASE_PDC_SSC   ((AT91PS_PDC)    0xFFFD4100) // (PDC_SSC) Base
   … Address
2200  #  define AT91C_BASE_SSC       ((AT91PS_SSC)    0xFFFD4000) // (SSC) Base
   … Address
2201  #  define AT91C_BASE_PDC_US1   ((AT91PS_PDC)    0xFFFC4100) // (PDC_US1) Base
   … Address
2202  #  define AT91C_BASE_US1       ((AT91PS_USART)  0xFFFC4000) // (US1) Base
   … Address
2203  #  define AT91C_BASE_PDC_US0   ((AT91PS_PDC)    0xFFFC0100) // (PDC_US0) Base
   … Address
2204  #  define AT91C_BASE_US0       ((AT91PS_USART)  0xFFFC0000) // (US0) Base
   … Address
2205  #  define AT91C_BASE_TWI       ((AT91PS_TWI)    0xFFFB8000) // (TWI) Base
   … Address
2206  #  define AT91C_BASE_TC2       ((AT91PS_TC)     0xFFFA0080) // (TC2) Base
   … Address
2207  #  define AT91C_BASE_TC1       ((AT91PS_TC)     0xFFFA0040) // (TC1) Base
   … Address
2208  #  define AT91C_BASE_TC0       ((AT91PS_TC)     0xFFFA0000) // (TC0) Base
   … Address
2209  #  define AT91C_BASE_TCB       ((AT91PS_TCB)    0xFFFA0000) // (TCB) Base
   … Address
2210  #  define AT91C_BASE_PWMC_CH3  ((AT91PS_PWMC_CH)    0xFFFCC260)
2211                                    // (PWMC_CH3) Base Address
2212  #  define AT91C_BASE_PWMC_CH2  ((AT91PS_PWMC_CH)    0xFFFCC240)
2213                                    // (PWMC_CH2) Base Address
2214  #  define AT91C_BASE_PWMC_CH1  ((AT91PS_PWMC_CH)    0xFFFCC220)
2215                                    // (PWMC_CH1) Base Address
2216  #  define AT91C_BASE_PWMC_CH0  ((AT91PS_PWMC_CH)    0xFFFCC200)
2217                                    // (PWMC_CH0) Base Address
2218  #  define AT91C_BASE_PWMC      ((AT91PS_PWMC)   0xFFFCC000) // (PWMC) Base
   … Address
2219  #  define AT91C_BASE_UDP       ((AT91PS_UDP)    0xFFFB0000) // (UDP) Base
   … Address
2220
2221  //
   … ********************************************************************************
```

```
2222 //               MEMORY MAPPING DEFINITIONS FOR AT91SAM7S64
2223 //
     ********************************************************************************
2224 #  define AT91C_ISRAM    ((char *)  0x00200000) // Internal SRAM base address
2225 #  define AT91C_ISRAM_SIZE   ((unsigned int) 0x00004000)    // Internal SRAM
     size in byte (16 Kbyte)
2226 #  define AT91C_IFLASH   ((char *)  0x00100000) // Internal ROM base address
2227 #  define AT91C_IFLASH_SIZE  ((unsigned int) 0x00010000)    // Internal ROM
     size in byte (64 Kbyte)
2228
2229 #endif
2230
```