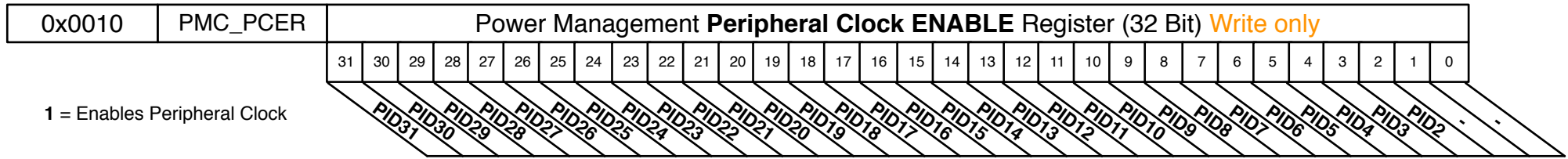
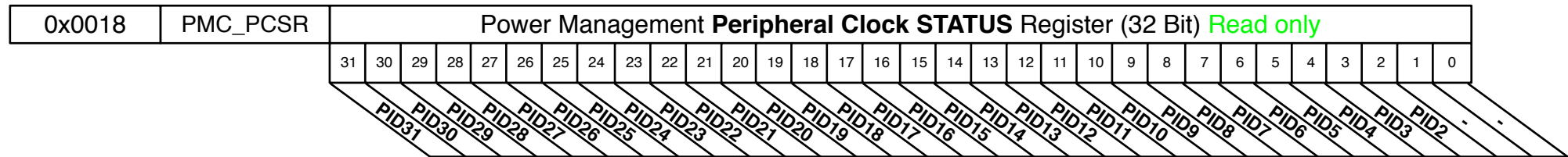
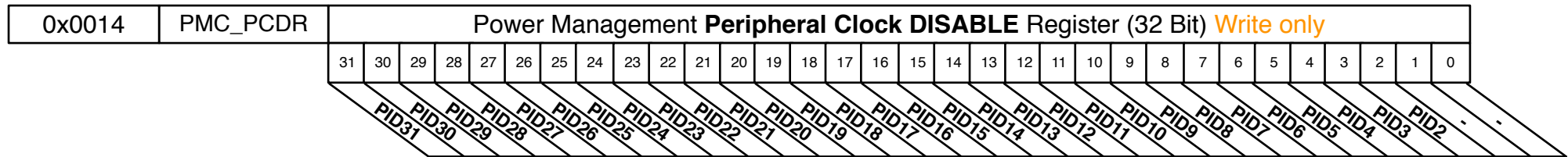


Using SPI:

1. Program PIO controller to assign SPI pins to their peripheral functions
2. Configure PMC to enable the SPI-Clock
- 3.1. Make sure the SPI_PCS (Chip Select) field of the SPI_MR (Mode) Register is set correctly
- 3.2. Make sure the SCBR (Baud Rate) field of the SPI_CSRx Register is set correctly, same for CPOL (Clock Polarity) and NCPHA (Clock Phase) which depend on the connected hardware...



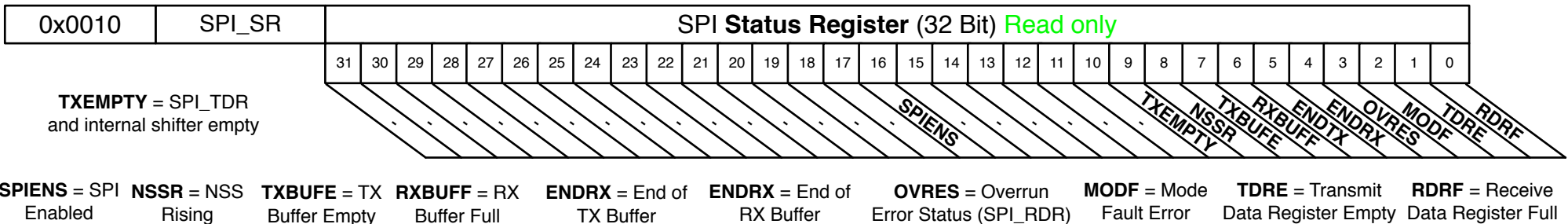
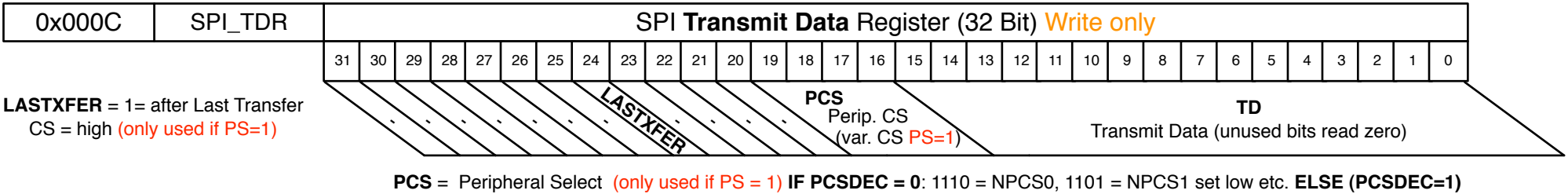
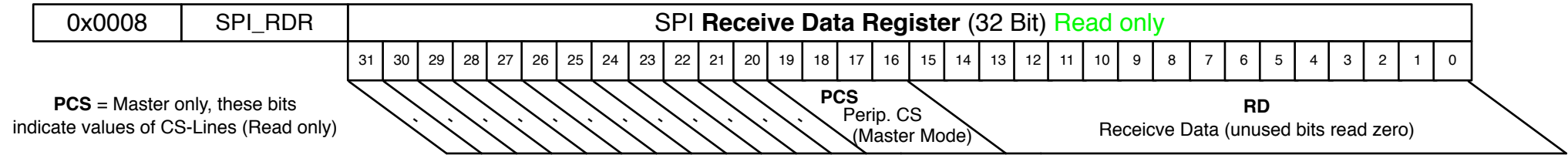
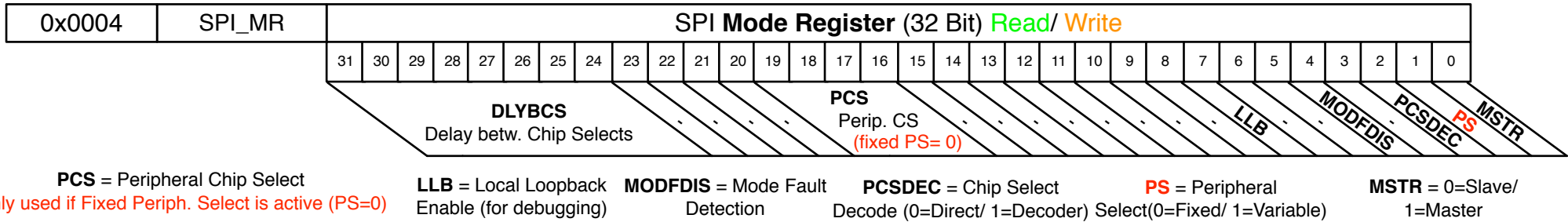
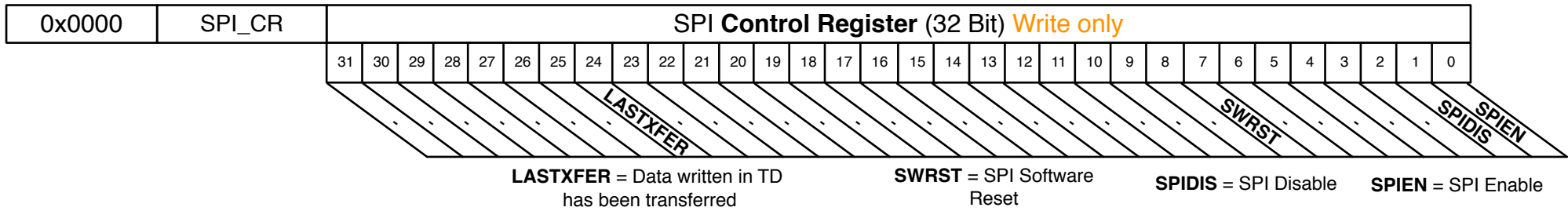
1 = Enables Peripheral Clock

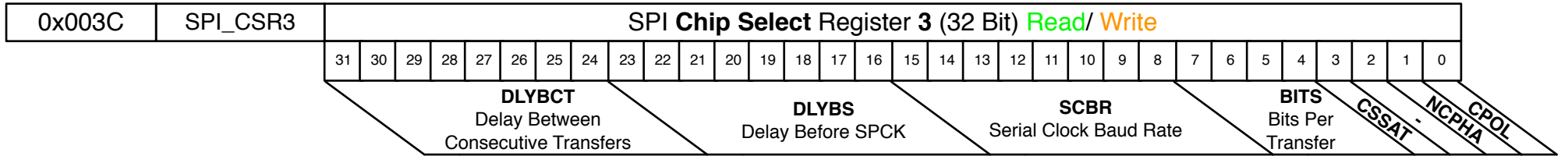
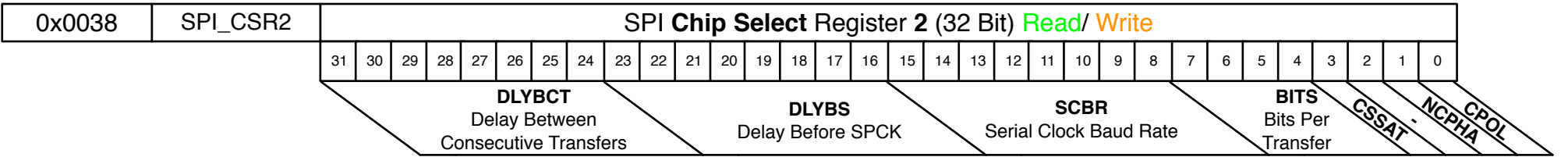
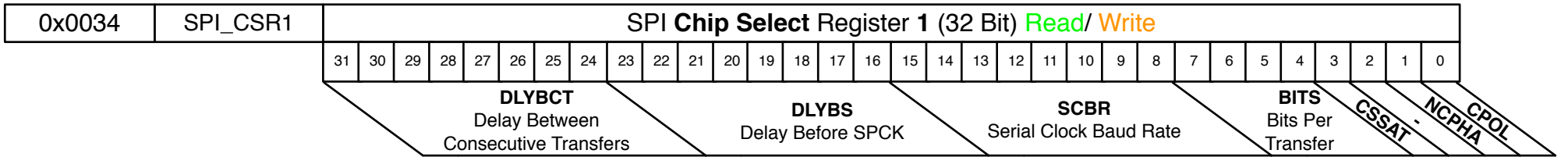
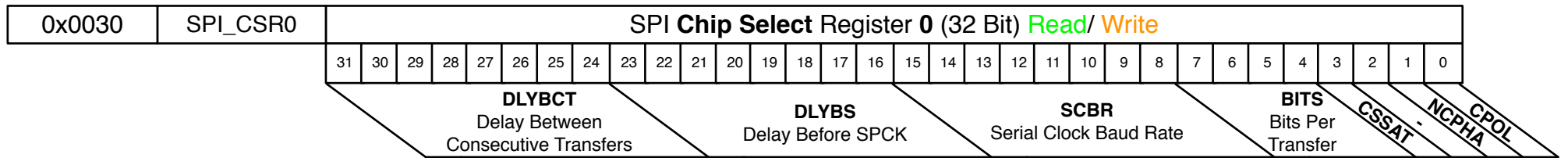


Peripheral ID	Peripheral Mnemonic	Peripheral Name
0	AIC	Advanced Interrupt Controller
1	SYSC	System Controller
2	PIOA	Parallel I/O Controller A
3	PIOB	Parallel I/O Controller B
4	SPI0	Serial Peripheral Interface 0
5	SPI1	Serial Peripheral Interface 1
6	US0	USART 0
7	US1	USART 1
8	SSC	Synchronous Serial Controller
9	TWI	Two-wire Interface
10	PWMC	Pulse Width Modulation Controller
11	UDP	USB Device Port
12	TC0	Timer/ Counter 0
13	TC1	Timer/ Counter 1
14	TC2	Timer/ Counter 2
15	CAN	CAN Controller

Peripheral ID	Peripheral Mnemonic	Peripheral Name
16	EMAC	Ethernet MAC
17	ADC	Analog-to Digital Converter
18	PIOA	Parallel I/O Controller A
19	Reserved	
20	Reserved	
21	Reserved	
22	Reserved	
23	Reserved	
24	Reserved	
25	Reserved	
26	Reserved	
27	Reserved	
28	Reserved	
29	Reserved	
30	AIC	Advanced Interrupt Controller
31	AIC	Advanced Interrupt Controller

Back





DLYBS (0-255)

$$\text{Delay} = 32 * \text{DLYBCT} / \text{MCK}$$

If DLBS == 0
Delay = SPCK/2

DLYBS (0-255)

$$\text{Delay} = \text{DLYBS} / \text{MCK}$$

If DLBS == 0
Delay = SPCK/2

SCBR (1-255)

$$\text{SPCK} = \text{MCK} / \text{SCBR}$$

BITS Per Transfer

- 0000 = 8 Bits
- 0001 = 9 Bits
- 0010 = 10 Bits
- 0011 = 11 Bits
- 0100 = 12 Bits
- 0101 = 13 Bits
- 0110 = 14 Bits
- 0111 = 15 Bits
- 1000 = 13 Bits

CPOL = Clock Polarity

NCPHA = Clock Phase

CSSAT = Chip Select
Active After Transfer

0 = Rises after Transf.
1 = Stays Low after Transf.

Back

		IO31	IO30	IO29	...	IO2	IO1	IO0
0x0000	PIO_PER	PIO Enable Register (32 Bit for 32 IO's)						
0x0004	PIO_PDR	PIO Disable Register (32 Bit for 32 IO's)						
0x0010	PIO_OER	Output Enable Register (32 Bit for 32 IO's)						
0x0014	PIO_ODR	Output Disable Register (32 Bit for 32 IO's)						
0x0020	PIO_IFER	Glitch Input Filter Enable Register (32 Bit for 32 IO's)						
0x0024	PIO_IFDR	Glitch Input Filter Disable Register (32 Bit for 32 IO's)						
0x0030	PIO_SODR	Set Output Data Register (32 Bit for 32 IO's)						
0x0034	PIO_CODR	Clear Output Data Register (32 Bit for 32 IO's)						
0x0040	PIO_IER	Interrupt Enable Register (32 Bit for 32 IO's)						
0x0044	PIO_IDR	Interrupt Disable Register (32 Bit for 32 IO's)						
0x0050	PIO_MDER	Multi Driver (Open Drain) Enable Register (32 Bit for 32 IO's)						
0x0054	PIO_MDDR	Multi Driver (Open Drain) Disable Register (32 Bit for 32 IO's)						
0x0060	PIO_PUDR	Pull-up Disable Register (32 Bit for 32 IO's)						
0x0064	PIO_PUER	Pull-up Enable Register (32 Bit for 32 IO's)						
0x0070	PIO_ASR	Peripheral A Select Register (32 Bit for 32 IO's)						
0x0074	PIO_BSR	Peripheral B Select Register (32 Bit for 32 IO's)						
0x00A0	PIO_OWER	Output Write Enable Register (32 Bit for 32 IO's)						
0x00A4	PIO_OWDR	Output Write Disable Register (32 Bit for 32 IO's)						

Back