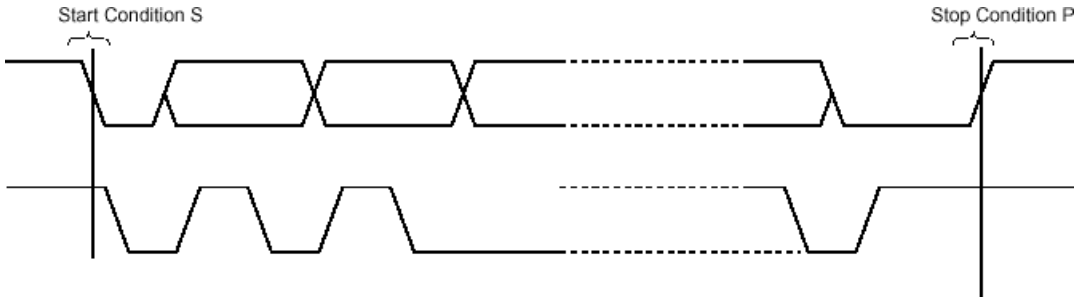
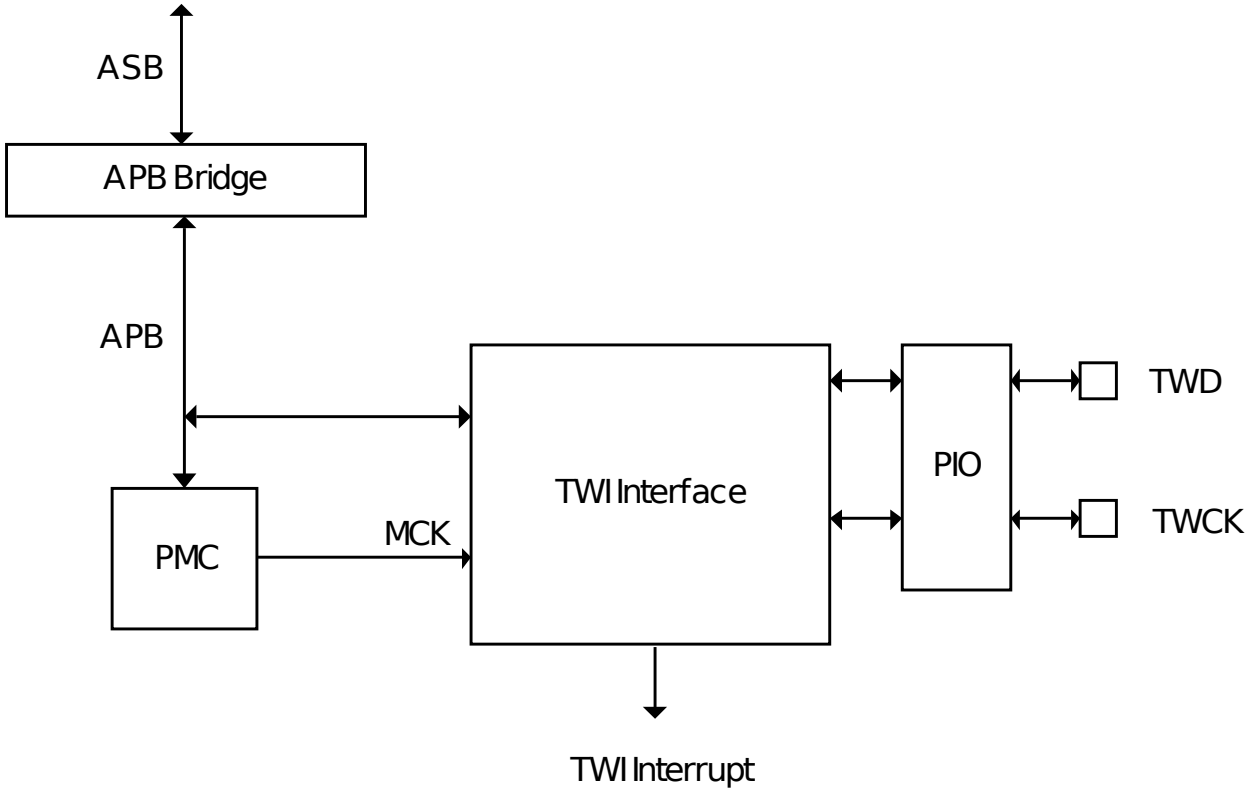
Two Wire Interface (TWI)



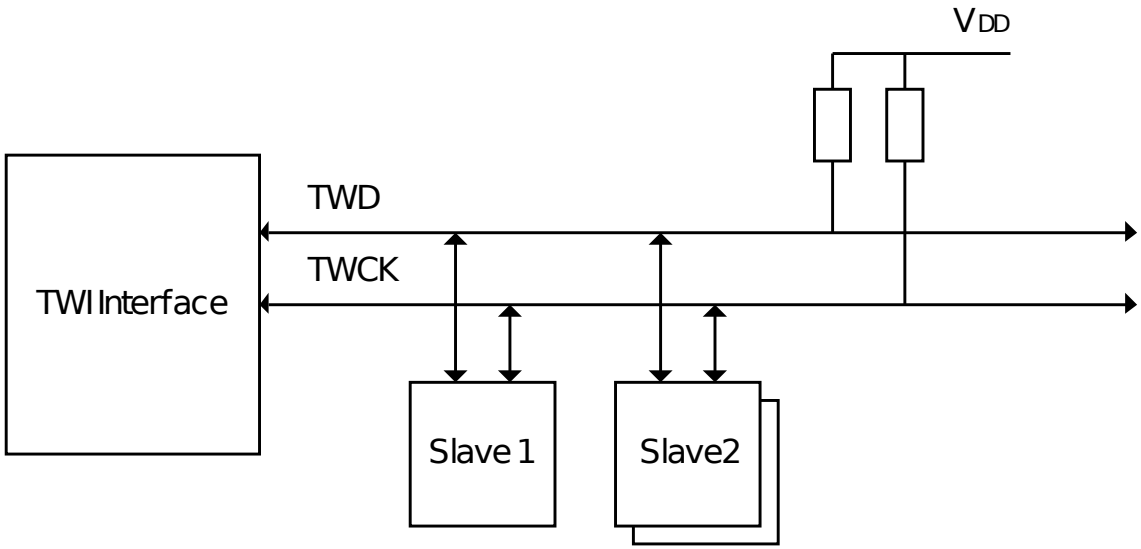
TWI Features

- Master mode support only
  - Master transmitter mode
  - Master receiver mode
  - All Two-wire Atmel EEPROMs Supported
- Programmable:
  - Clock baud rate Up to 400 Kbits
  - Up to Three bytes internal address (device up to16 Mbytes)
  - Support 7-bit and 10-bit addressing
- Support fast I^2C mode (up to 400kHz)

TWI Block Diagram



TWI Application



Dependencies

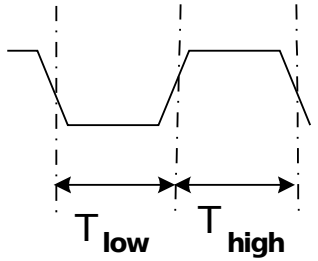
- PMC has to be programmed 1st for TWI to work
- PIO Controller has to be programmed for the pins to behave as intended
  - Dedicate the both as peripheral
  - Define the both line as open drain
- For example:
  - Configure TWI PIOs
  - Configure PMC by enabling TWI clock
  - Configure TWI in master mode
    - Disable interrupts
    - Reset peripheral
    - Set Master mode
  - Set TWI Clock Waveform Generator Register
    - CKDIV, CHDIV and CLDIV

TWI Clock Waveform Generator Register TWI_CWGR

18 CKDIV 16	15 CHDIV 8	7 CLDIV 0

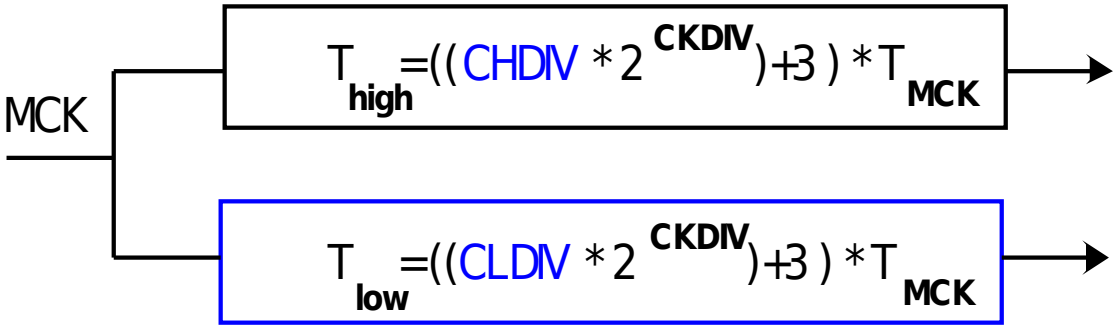
- CLDIV: The TWCK low period T_{low}
- CHDIV: The TWCK high period T_{high}

TWCK



- CKDIV: Clock divider
  - Increase the TWCK period

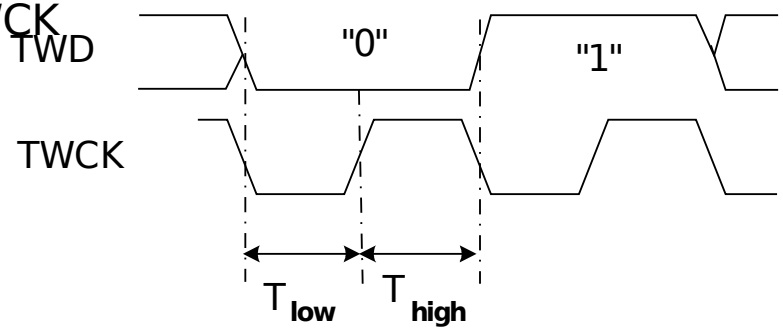
Clock Waveform Generator



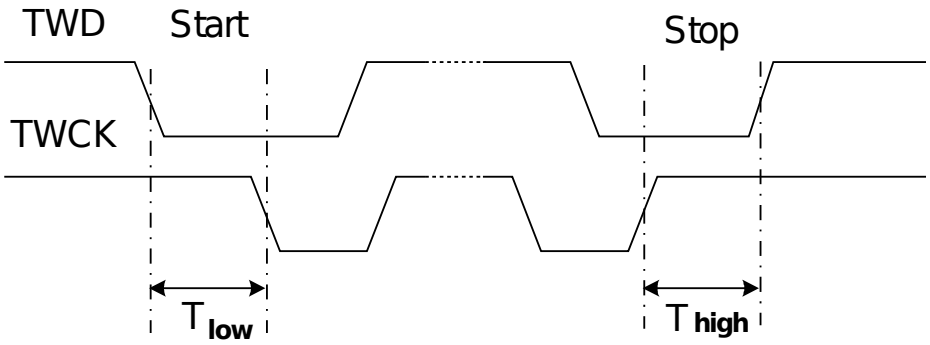
MCK	T_{MCK}	CKDIV	CHDIV		CLDIV				TWCK	
Hz	Second	0 to 7	0 to 255	$T_{high} = ((CHDIV * 2^{CKDIV}) + 3) T_{MCK}$	0 to 255	$T_{low} = ((CLDIV * 2^{CKDIV}) + 3) T_{MCK}$			Second	Hz
48 000 000	**20,8E-9**	2	15	**1,3E-6**	15	**1,3E-6**			2,6E-6	381,0E+3

Clock Generation

- Data are sampled on TWCK rising edge
  - Data Stable during high period of TWCK
  - Change during falling edge
  - Sampling on the rising edge



- Start and stop condition
  - TWD Falling edge while TWCK is high indicate **Start** Condition
  - TWD Rising edge while TWCK is high indicate **Stop** Condition



Control Register TWI_CR

- START Send a start
- STOP Send a stop after a complete transmission
- MSEN = 1 = Master Mode ENABLE
- MSDIS = 1 = Master Mode DISABLE

Both = Master mode = Disabled

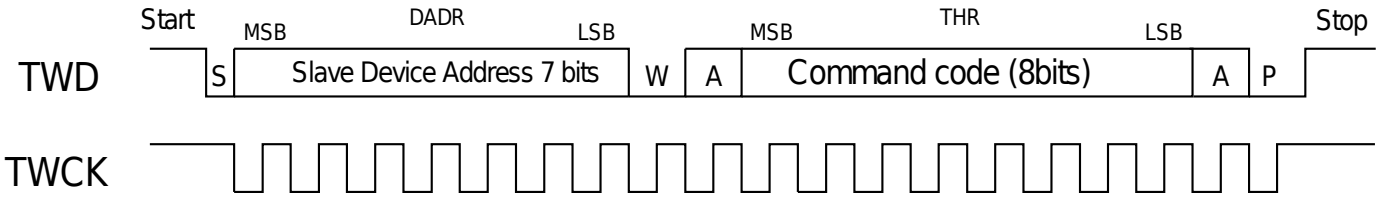
- SWRST = 1 = RESET = software controlled hardware reset
  - Writing a zero to this register has no effect
  - SWRST cleared by hardware

Master Mode Register TWI_MMR

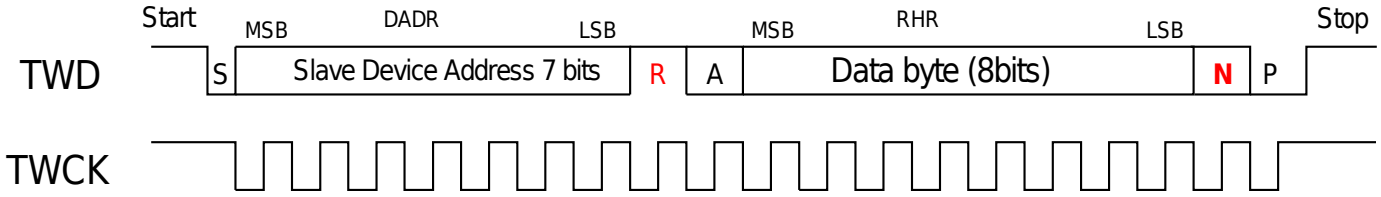
- DADR: Device bus address (0 to 127)
  - Used to access slave devices (Hard coded)
- MREAD: Master read direction
  - Master read direction
- IADRSZ: Internal Devices address size
  - 0 : No internal address (Send byte protocol)
  - 1: One-byte internal Device address size (0 to 256)
  - 2: Two-byte internal Device address size (0 to 65535)
  - 3: Three-byte internal Device address size (0 to 16 Mbytes)

Byte protocol Data Transfer

- Set the byte protocol excluding internal device address size
  - Master mode register IADRSZ = 0
- Write command



- Read Data from Slave



S: Start
W: Write
R: Read
A: Acknolewdge ACK
N: Not Acknolewdge NACK
DADR: Device address (Slave)
IADR: Internal device address

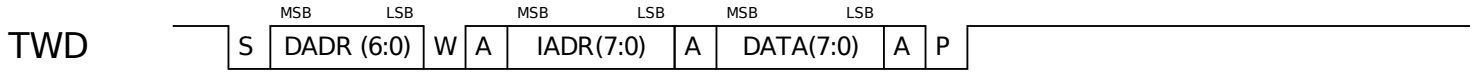
Internal Address Register TWI_IADR

- IADR: Internal device address (0 to 16 Mbytes)
  - Used to access slave devices internal mapping memory

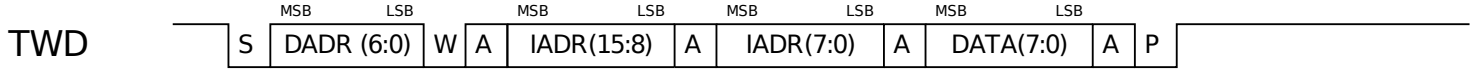
Write protocol Data Transfer

- Set the byte protocol including internal device address size
  - Master mode register IADRSZ # 0
  - Internal Address Register TWI_IADR
- Write command

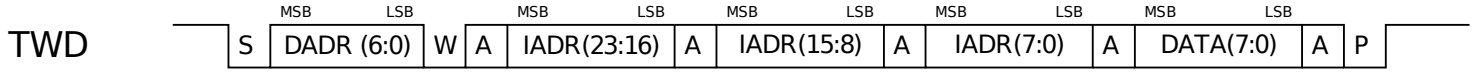
One byte internal Address



Two bytes internal Address



Three bytes internal Address

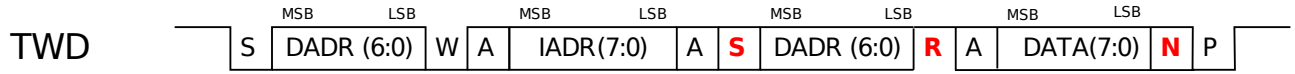


S: Start
W: Write
R: Read
A: Acknolewdge ACK
N: Not Acknolewdge NACK
DADR: Device address (Slave)
IADR: Internal device address

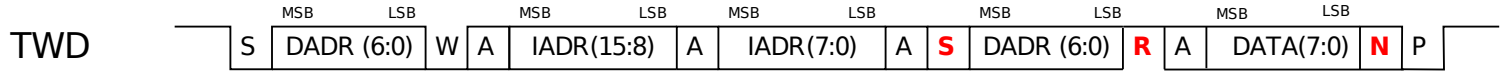
Read protocol Data Transfer

- Set the byte protocol including internal device address size
  - Master mode register IADRSZ # 0
  - Internal Address Register TWI_IADR
- Read command

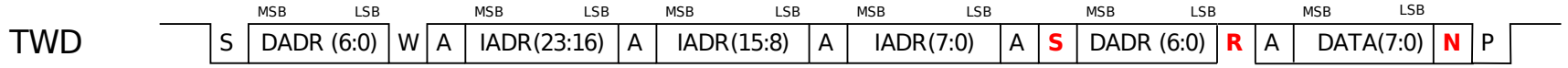
One byte internal Address



Two bytes internal Address



Three bytes internal Address



S: Start
W: Write
R: Read
A: Acknolewdge ACK
N: Not Acknolewdge NACK
DADR: Device address (Slave)
IADR: Internal device address

TWI Interrupts

- TWI Interrupt Enable Register TWI_IER (Write Only)
  - 0 = No effect
  - 1 = Enable
- TWI Interrupt Disable Register TWI_IDR (Write Only)
  - 0 = No effect
  - 1 = Disable
- TWI Interrupt Mask Register TWI_IMR (Read Only)
  - 0 = Not enabled
  - 1 = Enabled

TWI_IER, TWI_IDR, TWI_IMR

NACK 8	UNRE 7	OVER 6	5	4	3	TXRDY 2	RXRDY 1	TXCOMP 0

TWI Interrupts

- Transmission Completed
- Receive Holding Register Ready
- Transmit Holding Register Ready
- Overrun Error
- Underrun Error
- Not Acknowledge Error

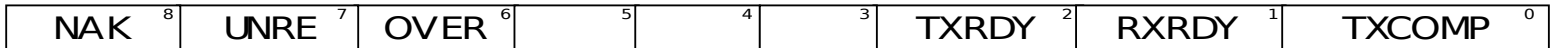
NAK 8	UNRE 7	OVER 6	5	4	3	TXRDY 2	RXRDY 1	TXCOMP 0

1 interrupt line goes to the AIC

Read TWI_SR to determine which interrupt occurred

TWI Status Register TWI_SR

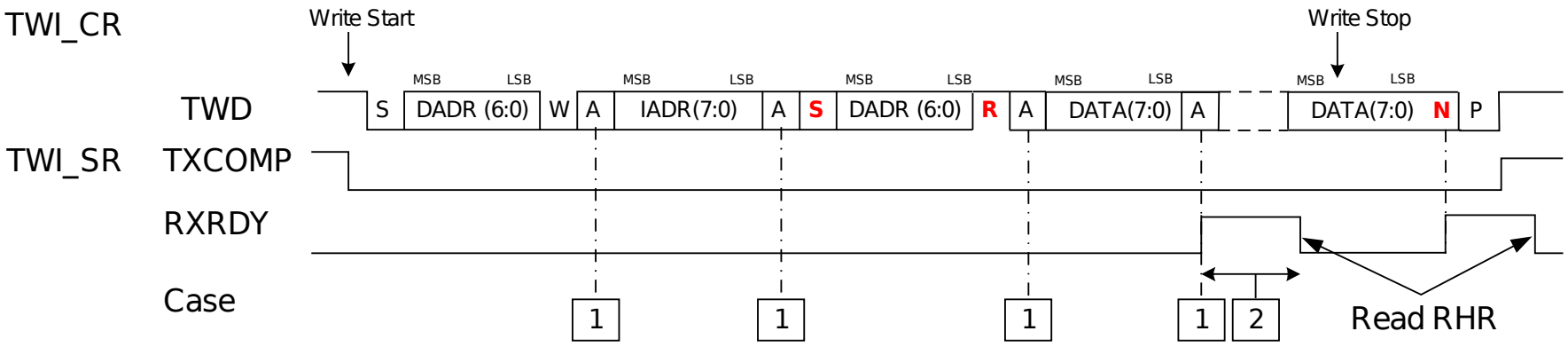
- Transmission Completed (Read And Write)
- Receive Holding Register Ready
  - 1: DATA byte has been received (Read mode)
- Transmit Holding Register Ready
  - 1: DATA byte must be transferred only (Write mode)
- Overrun Error
  - 1: Detect an overrun (Read mode)
- Underrun Error
  - 1: Detect an overrun (Write mode)
- Acknowledge
  - 1: Detect (Read & Write mode)



TWI Read Status bit

- Transmission Completed (0: during transmission)
- Receive Holding Register Ready (1: When the RHR register is full)
- Overrun Error (1: in case 2 if RXRDY=1 when other data received)
- Acknowledge (1: in case 1 if read NACK **N**)
  - if NACK stop the transmission and Send un Stop

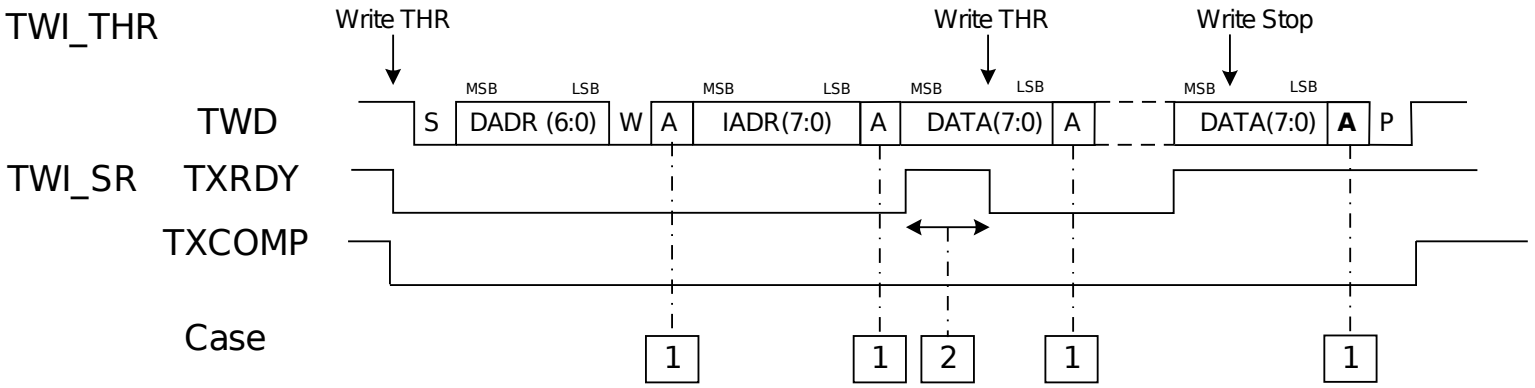
Read some data at one byte size internal Address



TWI Write Status bit

- Transmission Completed (0: during transmission)
- Transmit Holding Register Ready (1: When the THR register is Empty)
- Underrun Error (1: in case 2 if TXRDY=1 when new data must be transmit)
- Acknowledge (1: in case 1 if read NACK **N**)
  - if NACK stop the transmission and send a Stop

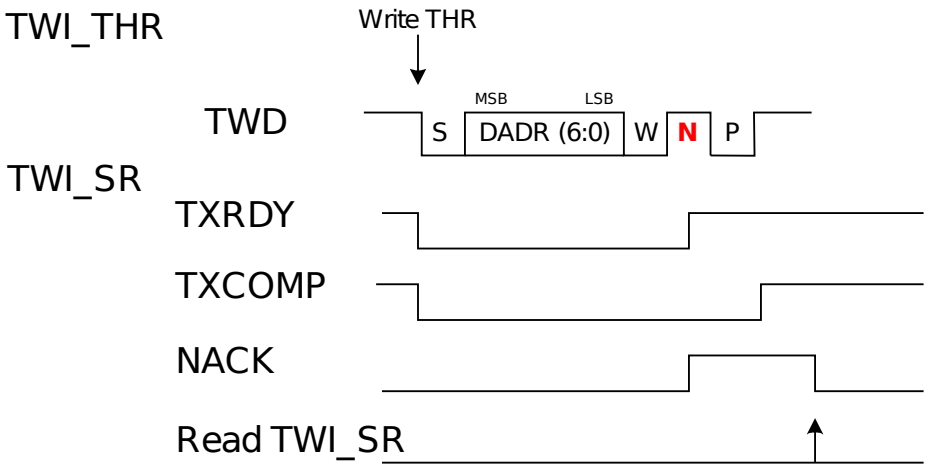
Write some data at one byte size internal Address



TWI Acknowledge Error

- Send a bad Device address
  - Write THR start transmission
  - Send Device Address (bad address)
  - NACK (**N**) detection
    - Set NACK (TWI_SR)
    - Set TXRDY (TWI_SR)
  - Send Stop
    - Set TXCOMP (TWI_SR)

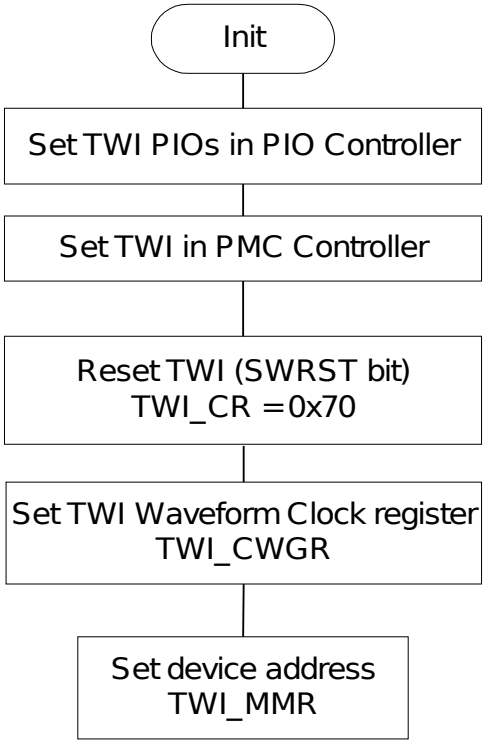
Write some data at one byte size internal Address



Software access

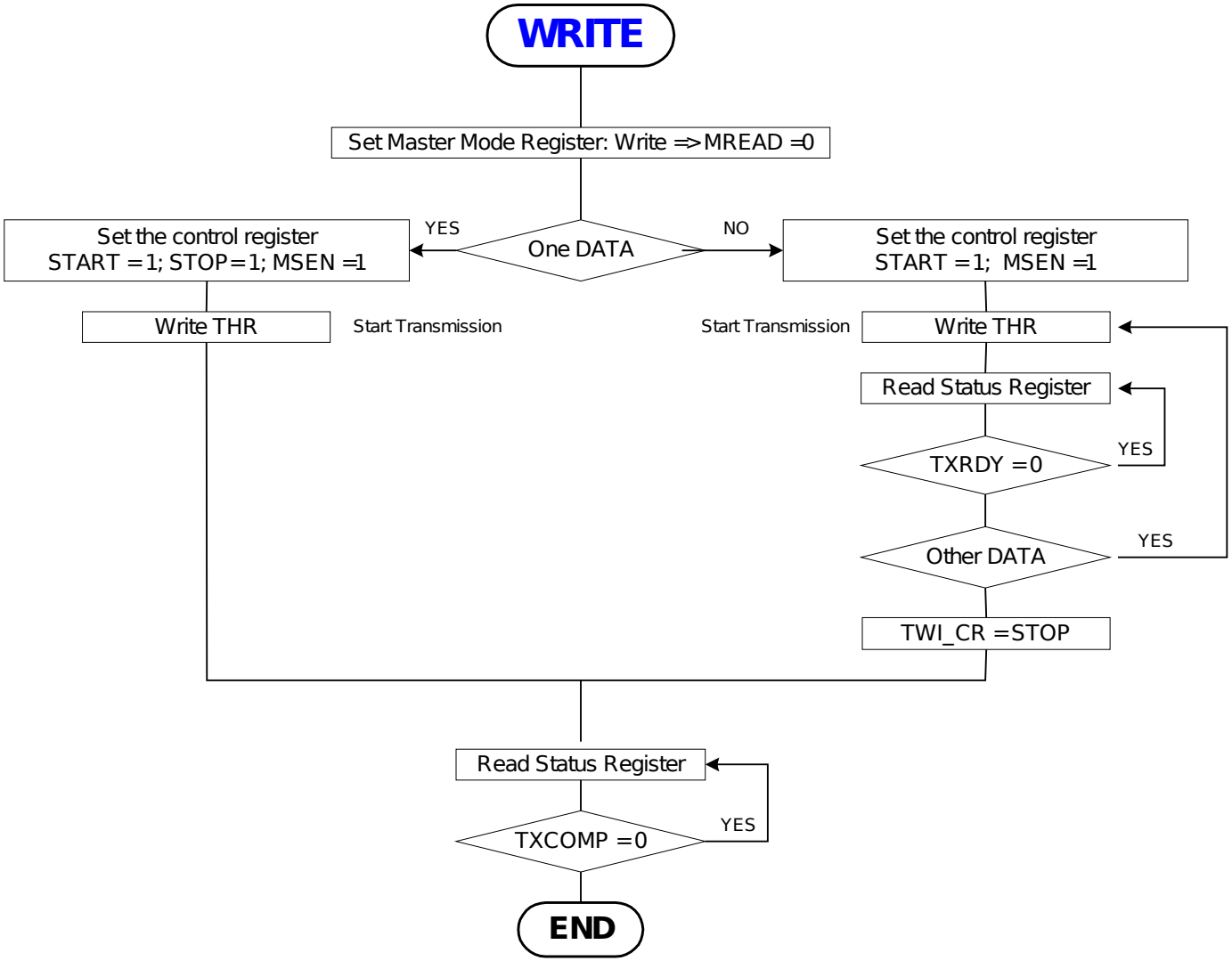
- All inline function
- Standard function is valid for AT91 core
- Register access function
  - AT91F_TWI_EnableIt
  - AT91F_TWI_DisableIt
  - AT91F_TWI_GetInterruptMaskStatus
  - AT91F_TWI_IsInterruptMasked
- **Setting function**
  - AT91F_TWI_CfgPIO
  - AT91F_TWI_CfgPMC
  - AT91F_TWI_Configure

Initialization



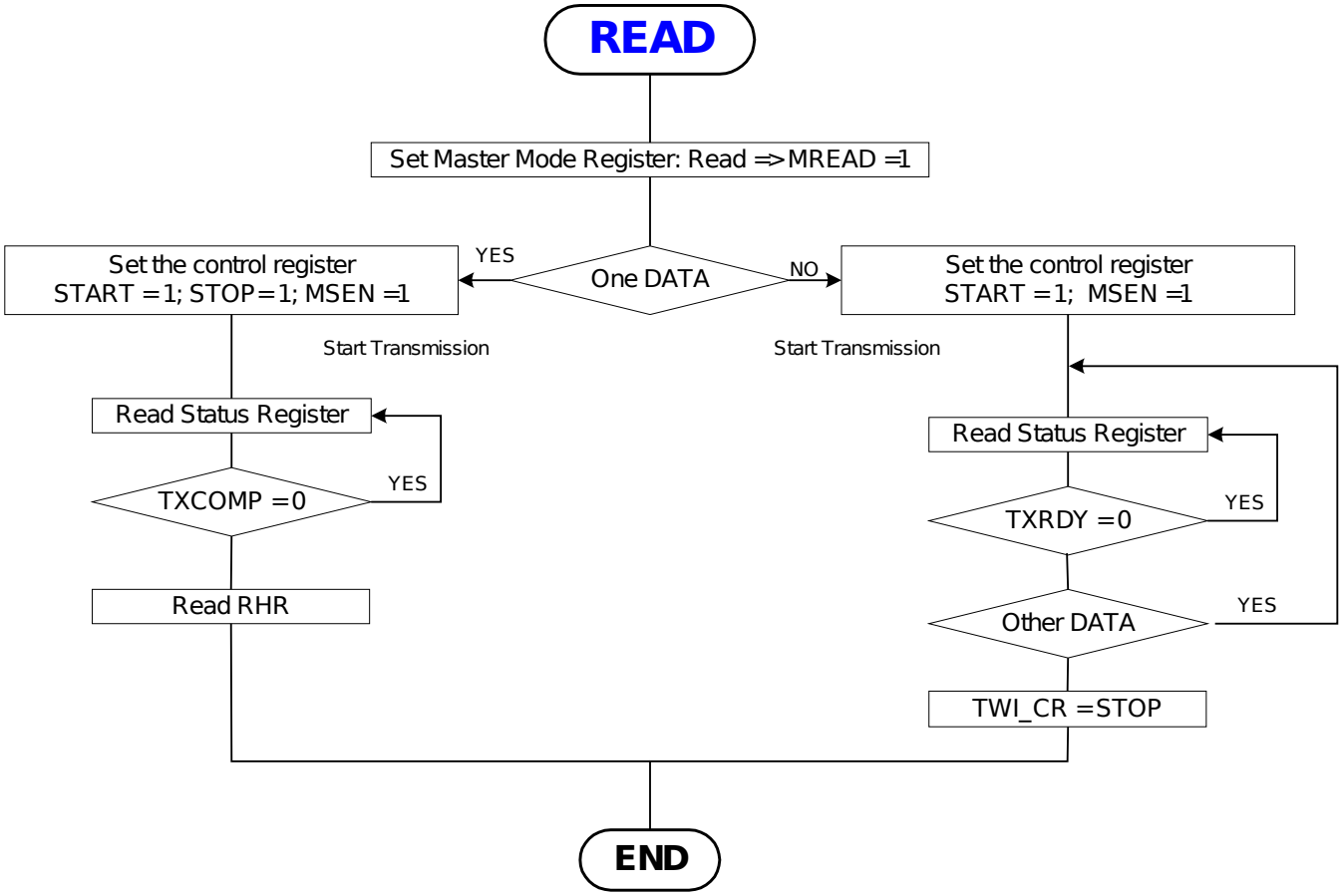
Initialization

- All inline function no cost for call
  - //* PIO AT91C_PA3_TWD & AT91C_PA4_TWCK (peripheral A)
    - AT91F_TWI_CfgPIO();
  - //* Power management
    - AT91F_TWI_CfgPMC();
  - //*TWI minimum Setting
    - AT91F_TWI_Configure(AT91C_BASE_TWI);
  - //* Set TWI Clock (MCK = 30MHz, TWI 8KHz) CKDIV=4, CHDIV=117, CLDIV=117)
    - *AT91C_TWI_CWGR= **0x047575**;
  - //*Set the device address 0x55 (7 bits), addressable space device 16 bits
    - *AT91C_TWI_MMR= **0x550200**;


WRITE
Set Master Mode Register: Write => MREAD =0
One DATA
YES
NO
Set the control register
START =1; STOP=1; MSEN =1
Set the control register
START =1; MSEN =1
Write THR
Start Transmission
Start Transmission
Write THR
Read Status Register
TXRDY =0
YES
Other DATA
YES
TWI_CR =STOP
Read Status Register
TXCOMP =0
YES
END

Write to Device

- Write value 0xAA to internal device address 0x00001
  - //* Set the Internal device address
    - *AT91C_TWI_IADR = 0x0001;
  - //* Set Write in mode register
    - *AT91C_TWI_MMR &= 0xFFFFEFFF;
  - //* Set control register
    - *AT91C_TWI_CR = AT91C_TWI_START | AT91C_TWI_MSEN | AT91C_TWI_STOP;
  - //* Set Data register for start transmission
    - *AT91C_TWI_THR = 0XAA;
  - //* Wait end transmission
    - Status = *AT91C_TWI_SR ;
    - while (!(status & AT91C_TWI_TXCOMP)){
    - Status = *AT91C_TWI_SR ; }


READ
Set Master Mode Register: Read =>MREAD =1
One DATA
YES
NO
Set the control register
START =1; STOP=1; MSEN =1
Set the control register
START =1; MSEN =1
Start Transmission
Start Transmission
Read Status Register
TXCOMP =0
YES
Read RHR
Read Status Register
TXRDY =0
YES
Other DATA
YES
TWI_CR =STOP
END

Read From Device

- Read data at internal device address 0x00001
  - //* Set the Internal device address
    - *AT91C_TWI_IADR = 0x0001;
  - //* Set Read in mode register
    - *AT91C_TWI_MMR |= 0x00001000;
  - //* Set control register and send start
    - *AT91C_TWI_CR = AT91C_TWI_START | AT91C_TWI_MSEN | AT91C_TWI_STOP;
  - //* Wait complete by TXCOMP or TXRDY
    - Status = *AT91C_TWI_SR ;
    - while (!(status & AT91C_TWI_TXCOMP)){
    - Status = *AT91C_TWI_SR ; }
  - //* Get data
    - Value = *AT91C_TWI_RHR;

TWI Summary

- High Speed. up to 400 K bits per second compatible Fast I2C
- Support Byte command & Internal device address protocol
- Individual Waveform clock Generator
- PIO Multiplex
- Error checking
  - Overrun, Underrun, NAK
- Master Mode only
- No Peripheral DMA support