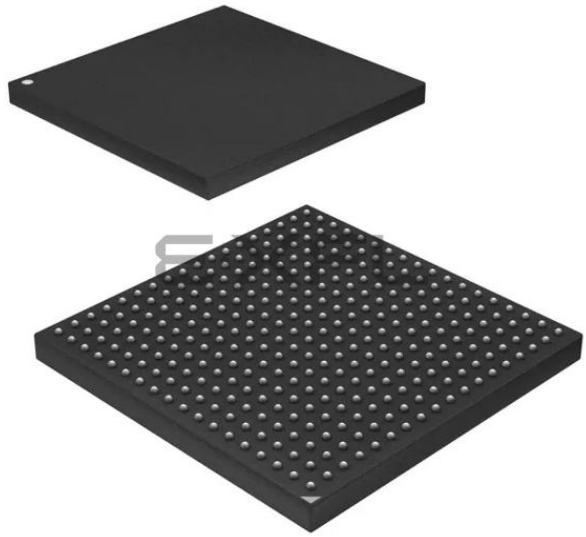


Welcome to E-XFL.COM

Understanding Embedded - Microprocessors

Embedded microprocessors are specialized computing chips designed to perform specific tasks within an embedded system. Unlike general-purpose microprocessors found in personal computers, embedded microprocessors are tailored for dedicated functions within larger systems, offering optimized performance, efficiency, and reliability. These microprocessors are integral to the operation of countless electronic devices, providing the computational power necessary for controlling processes, handling data, and managing communications.

Applications of Embedded - Microprocessors

Embedded microprocessors are utilized across a broad spectrum of applications, making them indispensable in

Details	
Product Status	Active
Core Processor	ARM926EJ-S
Number of Cores/Bus Width	1 Core, 32-Bit
Speed	400MHz
Co-Processors/DSP	-
RAM Controllers	LPDDR, LPSDR, DDR2, SDR, SRAM
Graphics Acceleration	No
Display & Interface Controllers	LCD, Touchscreen
Ethernet	10/100Mbps
SATA	-
USB	USB 2.0 (3)
Voltage - I/O	1.8V, 3.3V
Operating Temperature	-40°C ~ 85°C (TA)
Security Features	-
Package / Case	324-TFBGA
Supplier Device Package	324-TFBGA (15x15)
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/at91sam9g46b-cu

13.6.4 RTC Calendar Register

Name: RTC_CALR

Address: 0xFFFFFDBC

Access: Read/Write

31	30	29	28	27	26	25	24
–	–	DATE					

23	22	21	20	19	18	17	16
DAY			MONTH				

15	14	13	12	11	10	9	8
YEAR							

7	6	5	4	3	2	1	0
–	CENT						

- **CENT: Current Century**

The range that can be set is 19–20 (BCD).

The lowest four bits encode the units. The higher bits encode the tens.

- **YEAR: Current Year**

The range that can be set is 00–99 (BCD).

The lowest four bits encode the units. The higher bits encode the tens.

- **MONTH: Current Month**

The range that can be set is 01–12 (BCD).

The lowest four bits encode the units. The higher bits encode the tens.

- **DAY: Current Day in Current Week**

The range that can be set is 1–7 (BCD).

The coding of the number (which number represents which day) is user-defined as it has no effect on the date counter.

- **DATE: Current Day in Current Month**

The range that can be set is 01–31 (BCD).

The lowest four bits encode the units. The higher bits encode the tens.

All non-significant bits read zero.

13.6.7 RTC Status Register

Name: RTC_SR

Address: 0xFFFFFDC8

Access: Read-only

31	30	29	28	27	26	25	24
–	–	–	–	–	–	–	–
23	22	21	20	19	18	17	16
–	–	–	–	–	–	–	–
15	14	13	12	11	10	9	8
–	–	–	–	–	–	–	–
7	6	5	4	3	2	1	0
–	–	–	CALEV	TIMEV	SEC	ALARM	ACKUPD

- **ACKUPD: Acknowledge for Update**

0: Time and calendar registers cannot be updated.

1: Time and calendar registers can be updated.

- **ALARM: Alarm Flag**

0: No alarm matching condition occurred.

1: An alarm matching condition has occurred.

- **SEC: Second Event**

0: No second event has occurred since the last clear.

1: At least one second event has occurred since the last clear.

- **TIMEV: Time Event**

0: No time event has occurred since the last clear.

1: At least one time event has occurred since the last clear.

The time event is selected in the TIMEVSEL field in RTC_CTRL (Control Register) and can be any one of the following events: minute change, hour change, noon, midnight (day change).

- **CALEV: Calendar Event**

0: No calendar event has occurred since the last clear.

1: At least one calendar event has occurred since the last clear.

The calendar event is selected in the CALEVSEL field in RTC_CR and can be any one of the following events: week change, month change and year change.

20.8.3 Write Waveforms

The write protocol is similar to the read protocol. It is depicted in Figure 20-12. The write cycle starts with the address setting on the memory address bus.

20.8.3.1 NWE Waveforms

The NWE signal is characterized by a setup timing, a pulse width and a hold timing.

1. NWE_SETUP: the NWE setup time is defined as the setup of address and data before the NWE falling edge;
2. NWE_PULSE: The NWE pulse length is the time between NWE falling edge and NWE rising edge;
3. NWE_HOLD: The NWE hold time is defined as the hold time of address and data after the NWE rising edge.

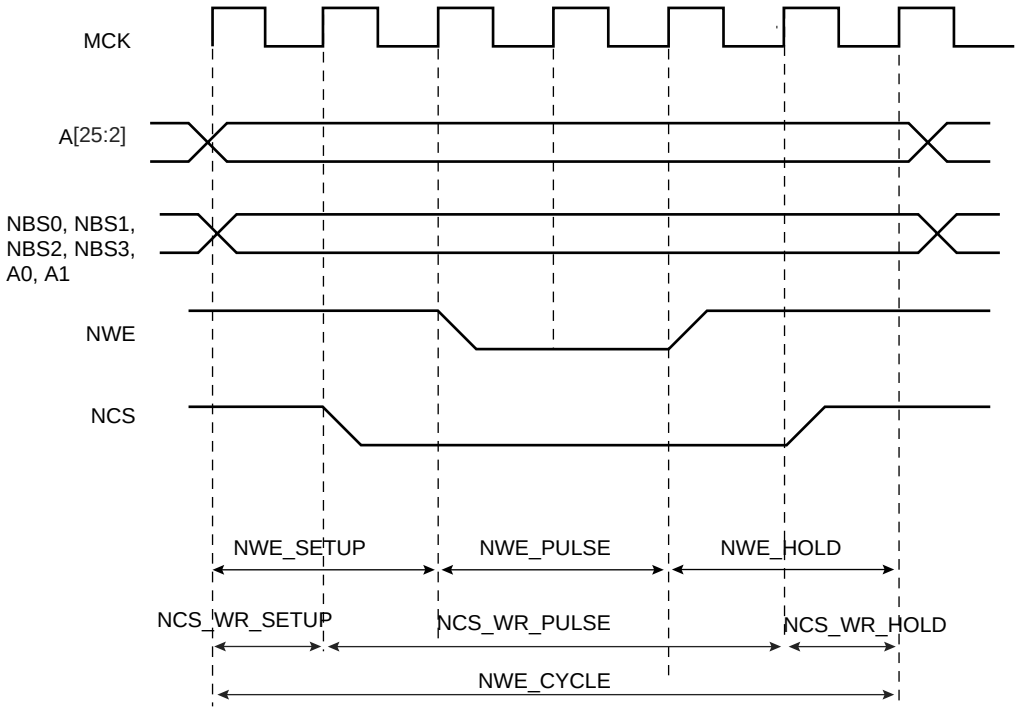
The NWE waveforms apply to all byte-write lines in Byte Write access mode: NWR0 to NWR3.

20.8.3.2 NCS Waveforms

The NCS signal waveforms in write operation are not the same that those applied in read operations, but are separately defined:

1. NCS_WR_SETUP: the NCS setup time is defined as the setup time of address before the NCS falling edge.
2. NCS_WR_PULSE: the NCS pulse length is the time between NCS falling edge and NCS rising edge;
3. NCS_WR_HOLD: the NCS hold time is defined as the hold time of address after the NCS rising edge.

Figure 20-12. Write Cycle



20.8.3.3 Write Cycle

The write_cycle time is defined as the total duration of the write cycle, that is, from the time where address is set on the address bus to the point where address may change. The total write cycle time is equal to:

NWE_CYCLE = NWE_SETUP + NWE_PULSE + NWE_HOLD

= NCS_WR_SETUP + NCS_WR_PULSE + NCS_WR_HOLD

20.15.4 SMC MODE Register

Name: SMC_MODE[0..5]

Address: 0xFFFFE80C [0], 0xFFFFE81C [1], 0xFFFFE82C [2], 0xFFFFE83C [3], 0xFFFFE84C [4], 0xFFFFE85C [5]

Access: Read/Write

31	30	29	28	27	26	25	24
–	–	PS		–	–	–	PMEN

23	22	21	20	19	18	17	16
–	–	–	TDF_MODE	TDF_CYCLES			

15	14	13	12	11	10	9	8
–	–	DBW		–	–	–	BAT

7	6	5	4	3	2	1	0
–	–	EXNW_MODE		–	–	WRITE_MODE	READ_MODE

- **READ_MODE:**

1: The read operation is controlled by the NRD signal.

– If TDF cycles are programmed, the external bus is marked busy after the rising edge of NRD.

– If TDF optimization is enabled (TDF_MODE =1), TDF wait states are inserted after the setup of NRD.

0: The read operation is controlled by the NCS signal.

– If TDF cycles are programmed, the external bus is marked busy after the rising edge of NCS.

– If TDF optimization is enabled (TDF_MODE =1), TDF wait states are inserted after the setup of NCS.

- **WRITE_MODE**

1: The write operation is controlled by the NWE signal.

– If TDF optimization is enabled (TDF_MODE =1), TDF wait states will be inserted after the setup of NWE.

0: The write operation is controlled by the NCS signal.

– If TDF optimization is enabled (TDF_MODE =1), TDF wait states will be inserted after the setup of NCS.

- **EXNW_MODE: NWAIT Mode**

The NWAIT signal is used to extend the current read or write signal. It is only taken into account during the pulse phase of the read and write controlling signal. When the use of NWAIT is enabled, at least one cycle hold duration must be programmed for the read and write controlling signal.

EXNW_MODE		NWAIT Mode
0	0	Disabled
0	1	Reserved
1	0	Frozen Mode
1	1	Ready Mode

- Disabled Mode: The NWAIT input signal is ignored on the corresponding Chip Select.
- Frozen Mode: If asserted, the NWAIT signal freezes the current read or write cycle. After deassertion, the read/write cycle is resumed from the point where it was stopped.
- Ready Mode: The NWAIT signal indicates the availability of the external device at the end of the pulse of the controlling read or write signal, to complete the access. If high, the access normally completes. If low, the access is extended until NWAIT returns high.

22.6.6 ECC Parity Register 5

Name: ECC_PR5

Access: Read-only

31	30	29	28	27	26	25	24
–	–	–	–	–	–	–	–
23	22	21	20	19	18	17	16
NPARITY5							
15	14	13	12	11	10	9	8
NPARITY5				WORDADDR5			
7	6	5	4	3	2	1	0
WORDADDR5					BITADDR5		

Once the entire main area of a page is written with data, the register content must be stored at any free location of the spare area.

- **BITADDR5: corrupted Bit Address in the page between the 2560th and the 3071st bytes**

During a page read, this value contains the corrupted bit offset where an error occurred, if a single error was detected. If multiple errors were detected, this value is meaningless.

- **WORDADDR5: corrupted Word Address in the page between the 2560th and the 3071st bytes**

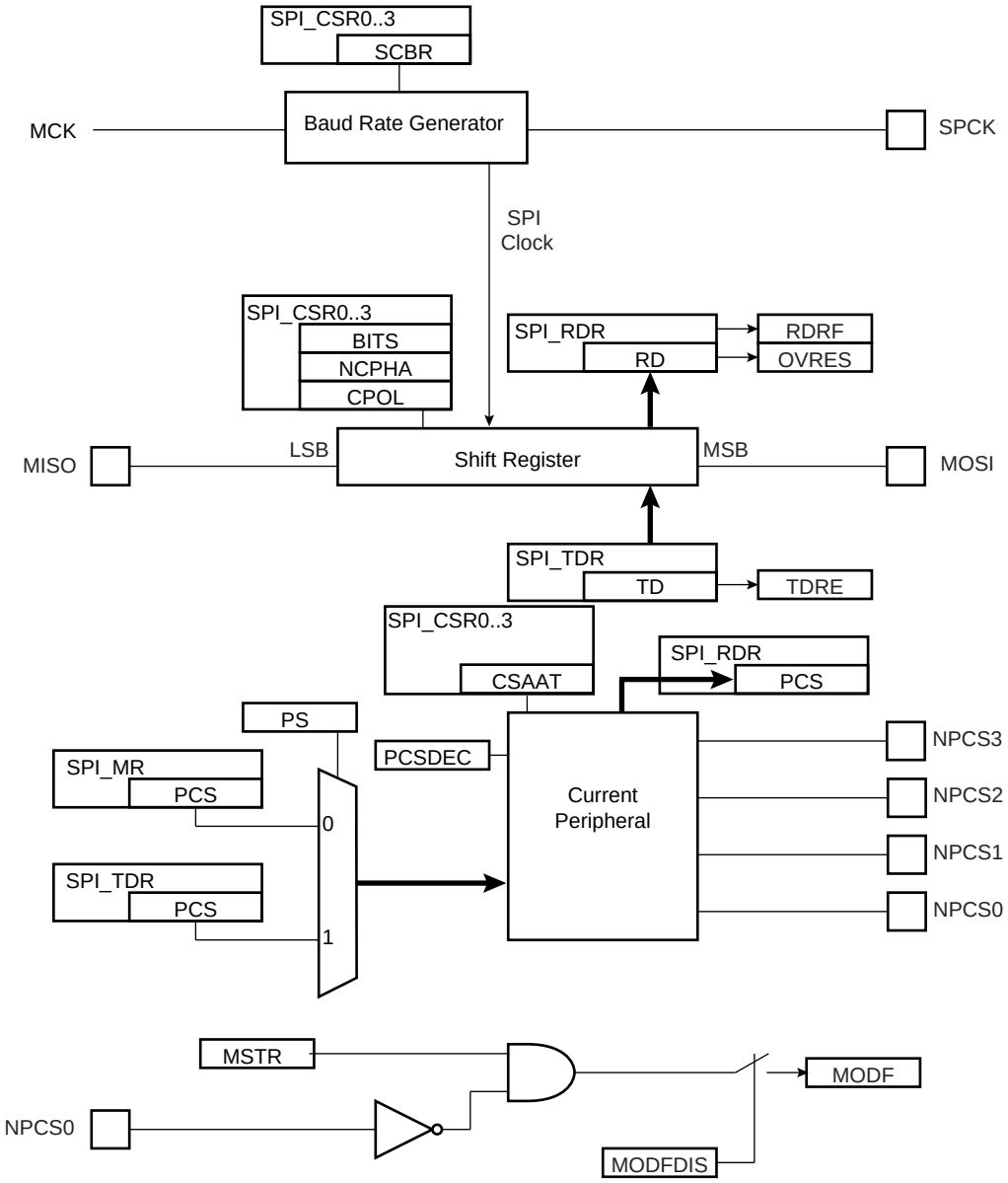
During a page read, this value contains the word address (8-bit word) where an error occurred, if a single error was detected. If multiple errors were detected, this value is meaningless.

- **NPARITY5**

Parity N

28.7.3.1 Master Mode Block Diagram

Figure 28-6. Master Mode Block Diagram



The Variable Peripheral Selection allows buffer transfers with multiple peripherals without reprogramming the Mode Register. Data written in SPI_TDR is 32 bits wide and defines the real data to be transmitted and the peripheral it is destined to. Using the PDC in this mode requires 32-bit wide buffers, with the data in the LSBs and the PCS and LASTXFER fields in the MSBs, however the SPI still controls the number of bits (8 to16) to be transferred through MISO and MOSI lines with the chip select configuration registers. This is not the optimal means in term of memory size for the buffers, but it provides a very effective means to exchange data with several peripherals without any intervention of the processor.

28.7.3.7 Transfer Size

Depending on the data size to transmit, from 8 to 16 bits, the PDC manages automatically the type of pointer's size it has to point to. The PDC will perform the following transfer size depending on the mode and number of bits per data.

Fixed Mode:

- 8-bit Data:
  Byte transfer,
  PDC Pointer Address = Address + 1 byte,
  PDC Counter = Counter - 1
- 8-bit to 16-bit Data:
  2 bytes transfer. n-bit data transfer with don’t care data (MSB) filled with 0’s,
  PDC Pointer Address = Address + 2 bytes,
  PDC Counter = Counter - 1

Variable Mode:

In variable Mode, PDC Pointer Address = Address +4 bytes and PDC Counter = Counter - 1 for 8 to 16-bit transfer size. When using the PDC, the TDRE and RDRF flags are handled by the PDC, thus the user’s application does not have to check those bits. Only End of RX Buffer (ENDRX), End of TX Buffer (ENDTX), Buffer Full (RXBUFF), TX Buffer Empty (TXBUFE) are significant. For further details about the Peripheral DMA Controller and user interface, refer to the PDC section of the product datasheet.

28.7.3.8 SPI Direct Access Memory Controller (DMAC)

In both fixed and variable mode the Direct Memory Access Controller (DMAC) can be used to reduce processor overhead.

The Fixed Peripheral Selection allows buffer transfers with a single peripheral. Using the DMAC is an optimal means, as the size of the data transfer between the memory and the SPI is either 8 bits or 16 bits. However, changing the peripheral selection requires the Mode Register to be reprogrammed.

The Variable Peripheral Selection allows buffer transfers with multiple peripherals without reprogramming the Mode Register. Data written in SPI_TDR is 32 bits wide and defines the real data to be transmitted and the peripheral it is destined to. Using the DMAC in this mode requires 32-bit wide buffers, with the data in the LSBs and the PCS and LASTXFER fields in the MSBs, however the SPI still controls the number of bits (8 to16) to be transferred through MISO and MOSI lines with the chip select configuration registers. This is not the optimal means in term of memory size for the buffers, but it provides a very effective means to exchange data with several peripherals without any intervention of the processor.

28.7.3.9 Peripheral Chip Select Decoding

The user can program the SPI to operate with up to 15 peripherals by decoding the four Chip Select lines, NPCS0 to NPCS3 with 1 of up to 16 decoder/demultiplexer. This can be enabled by writing the PCSDEC bit at 1 in the Mode Register (SPI_MR).

When operating without decoding, the SPI makes sure that in any case only one chip select line is activated, i.e., one NPCS line driven low at a time. If two bits are defined low in a PCS field, only the lowest numbered chip select is driven low.

28.8.2 SPI Mode Register

Name: SPI_MR

Address: 0xFFFA4004 (0), 0xFFFA8004 (1)

Access: Read/Write

31	30	29	28	27	26	25	24
DLYBCS							
23	22	21	20	19	18	17	16
–	–	–	–	PCS			
15	14	13	12	11	10	9	8
–	–	–	–	–	–	–	–
7	6	5	4	3	2	1	0
LLB	–	–	MODFDIS	–	PCSDEC	PS	MSTR

- **MSTR: Master/Slave Mode**

0: SPI is in Slave mode.

1: SPI is in Master mode.

- **PS: Peripheral Select**

0: Fixed Peripheral Select.

1: Variable Peripheral Select.

- **PCSDEC: Chip Select Decode**

0: The chip selects are directly connected to a peripheral device.

1: The four chip select lines are connected to a 4- to 16-bit decoder.

When PCSDEC equals one, up to 15 Chip Select signals can be generated with the four lines using an external 4- to 16-bit decoder. The Chip Select Registers define the characteristics of the 15 chip selects according to the following rules:

SPI_CSR0 defines peripheral chip select signals 0 to 3.

SPI_CSR1 defines peripheral chip select signals 4 to 7.

SPI_CSR2 defines peripheral chip select signals 8 to 11.

SPI_CSR3 defines peripheral chip select signals 12 to 14.

- **MODFDIS: Mode Fault Detection**

0: Mode fault detection is enabled.

1: Mode fault detection is disabled.

- **WDRBT: Wait Data Read Before Transfer**

0: No effect. In master mode, a transfer can be initiated whatever the state of the Receive Data Register is.

1: In Master Mode, a transfer can start only if the Receive Data Register is empty, i.e. does not contain any unread data. This mode prevents overrun error in reception.

29.6.14 PIO Interrupt Enable Register

Name: PIO_IER

Address: 0xFFFFF240 (PIOA), 0xFFFFF440 (PIOB), 0xFFFFF640 (PIOC), 0xFFFFF840 (PIOD), 0xFFFFFA40 (PIOE)

Access: Write-only

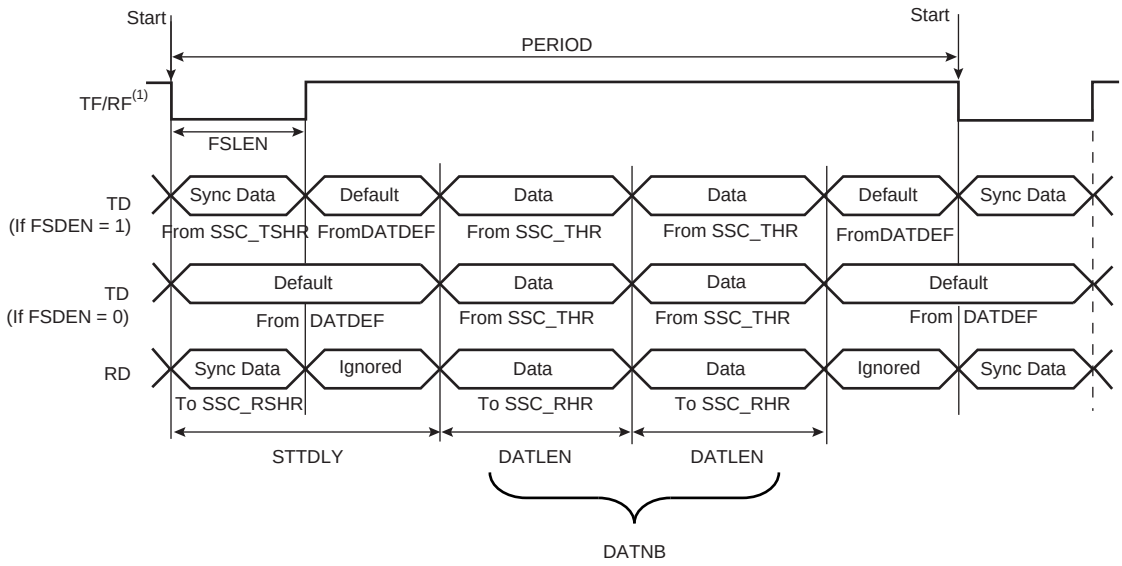
31	30	29	28	27	26	25	24
P31	P30	P29	P28	P27	P26	P25	P24
23	22	21	20	19	18	17	16
P23	P22	P21	P20	P19	P18	P17	P16
15	14	13	12	11	10	9	8
P15	P14	P13	P12	P11	P10	P9	P8
7	6	5	4	3	2	1	0
P7	P6	P5	P4	P3	P2	P1	P0

- **P0–P31: Input Change Interrupt Enable**

0: No effect.

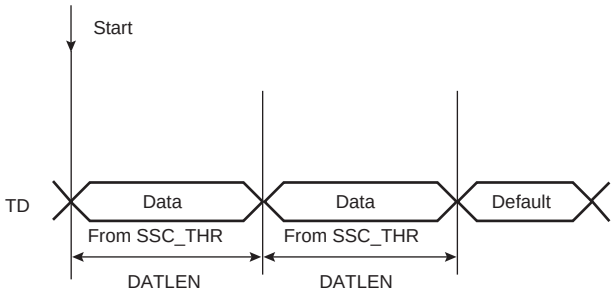
1: Enables the Input Change Interrupt on the I/O line.

Figure 33-14. Transmit and Receive Frame Format in Edge/Pulse Start Modes



Note: 1. Example of input on falling edge of TF/RF.

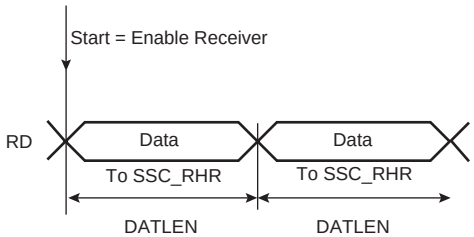
Figure 33-15. Transmit Frame Format in Continuous Mode



Start: 1. TXEMPTY set to 1
2. Write into the SSC_THR

Note: 1. STTDLY is set to 0. In this example, SSC_THR is loaded twice. FSDEN value has no effect on the transmission. SyncData cannot be output in continuous mode.

Figure 33-16. Receive Frame Format in Continuous Mode



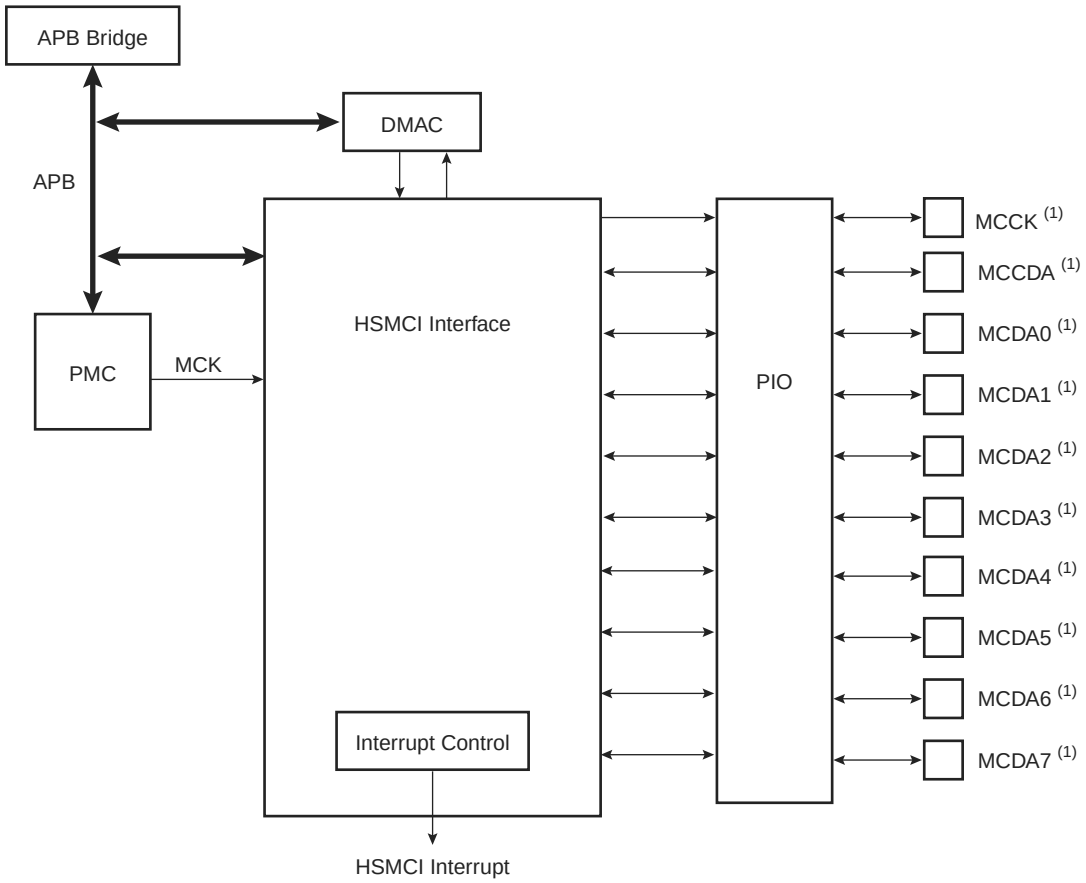
Note: 1. STTDLY is set to 0.

33.7.8 Loop Mode

The receiver can be programmed to receive transmissions from the transmitter. This is done by setting the Loop Mode (LOOP) bit in SSC_RFMR. In this case, RD is connected to TD, RF is connected to TF and RK is connected to TK.

35.3 Block Diagram

Figure 35-1. Block Diagram



Notes: 1. When several HSMCI (x HSMCI) are embedded in a product, MCCK refers to HSMCIx_CK, MCCDA to HSMCIx_CDA, MCDAy to HSMCIx_DAy.

35.14.2 HSMCI Mode Register

Name: HSMCI_MR

Address: 0xFFF80004 (0), 0xFFFD0004 (1)

Access: Read/Write

31	30	29	28	27	26	25	24
BLKLEN							
23	22	21	20	19	18	17	16
BLKLEN							
15	14	13	12	11	10	9	8
–	PADV	FBYTE	WRPROOF	RDPROOF	PWSDIV		
7	6	5	4	3	2	1	0
CLKDIV							

This register can only be written if the WPEN bit is cleared in “HSMCI Write Protect Mode Register” on page 819.

- **CLKDIV: Clock Divider**

High Speed MultiMedia Card Interface clock (MCCK or HSMCI_CK) is Master Clock (MCK) divided by (2*(CLKDIV+1)).

- **PWSDIV: Power Saving Divider**

High Speed MultiMedia Card Interface clock is divided by $2^{(PWSDIV)}$ + 1 when entering Power Saving Mode.

Warning: This value must be different from 0 before enabling the Power Save Mode in the HSMCI_CR (PWSEN bit).

- **RDPROOF Read Proof Enable**

Enabling Read Proof allows to stop the HSMCI Clock during read access if the internal FIFO is full. This will guarantee data integrity, not bandwidth.

0: Disables Read Proof.

1: Enables Read Proof.

- **WRPROOF Write Proof Enable**

Enabling Write Proof allows to stop the HSMCI Clock during write access if the internal FIFO is full. This will guarantee data integrity, not bandwidth.

0: Disables Write Proof.

1: Enables Write Proof.

- **FBYTE: Force Byte Transfer**

Enabling Force Byte Transfer allow byte transfers, so that transfer of blocks with a size different from modulo 4 can be supported.

Warning: BLKLEN value depends on FBYTE.

0: Disables Force Byte Transfer.

1: Enables Force Byte Transfer.

37.6.12 UDPHS Endpoint Set Status Register

Name: UDPHS_EPTSETSTAx [x=0..6]

Address: 0xFFF78114 [0], 0xFFF78134 [1], 0xFFF78154 [2], 0xFFF78174 [3], 0xFFF78194 [4], 0xFFF781B4 [5], 0xFFF781D4 [6]

Access: Write-only

31	30	29	28	27	26	25	24
–	–	–	–	–	–	–	–
23	22	21	20	19	18	17	16
–	–	–	–	–	–	–	–
15	14	13	12	11	10	9	8
–	–	–	–	TX_PK_RDY	–	KILL_BANK	–
7	6	5	4	3	2	1	0
–	–	FRCESTALL	–	–	–	–	–

- **FRCESTALL: Stall Handshake Request Set**

0: No effect.

1: Set this bit to request a STALL answer to the host for the next handshake

Refer to chapters 8.4.5 (Handshake Packets) and 9.4.5 (Get Status) of the *Universal Serial Bus Specification, Rev 2.0* for more information on the STALL handshake.

- **KILL_BANK: KILL Bank Set (for IN Endpoint)**

0: No effect.

1: Kill the last written bank.

- **TX_PK_RDY: TX Packet Ready Set**

0: No effect.

1: Set this bit after a packet has been written into the endpoint FIFO for IN data transfers

  – This flag is used to generate a Data IN transaction (device to host).
  – Device firmware checks that it can write a data payload in the FIFO, checking that TX_PK_RDY is cleared.
  – Transfer to the FIFO is done by writing in the “Buffer Address” register.
  – Once the data payload has been transferred to the FIFO, the firmware notifies the UDPHS device setting TX_PK_RDY to one.
  – UDPHS bus transactions can start.
  – TXCOMP is set once the data payload has been received by the host.
  – Data should be written into the endpoint FIFO only after this bit has been cleared.
  – Set this bit without writing data to the endpoint FIFO to send a Zero Length Packet.

39.7.3 Pressure Measurement Method

The method to measure the pressure (Rp) applied to the touchscreen is based on the knowledge of the X-Panel resistance (Rxp).

Three conversions (Xpos,Z1,Z2) are necessary to determine the value of Rp (Zaxis resistance).

Rp = Rxp*(Xpos/1024)*[(Z2/Z1)-1]

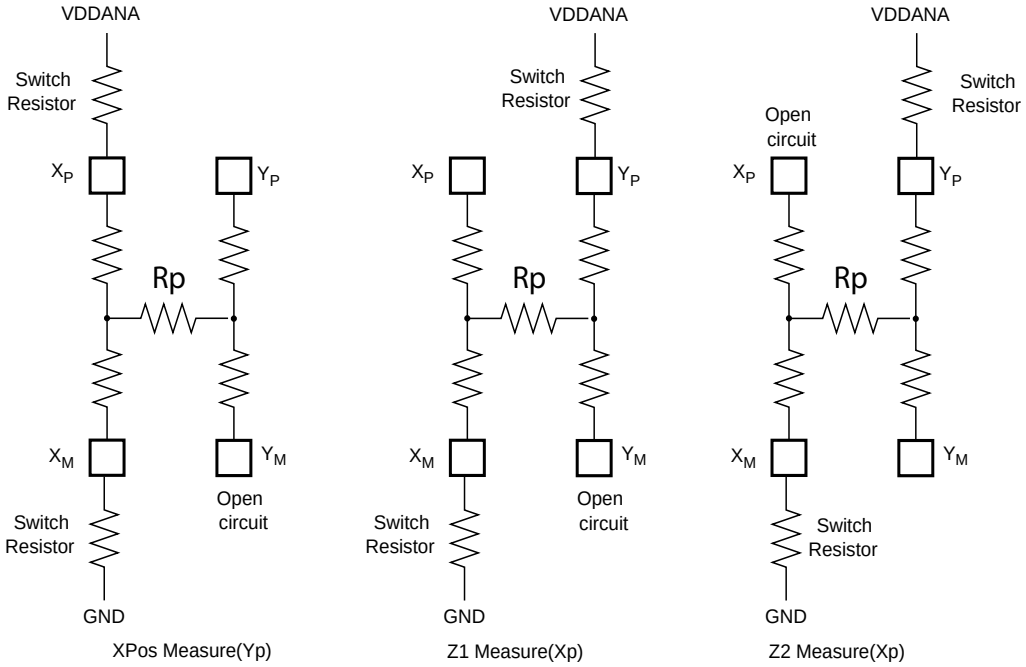
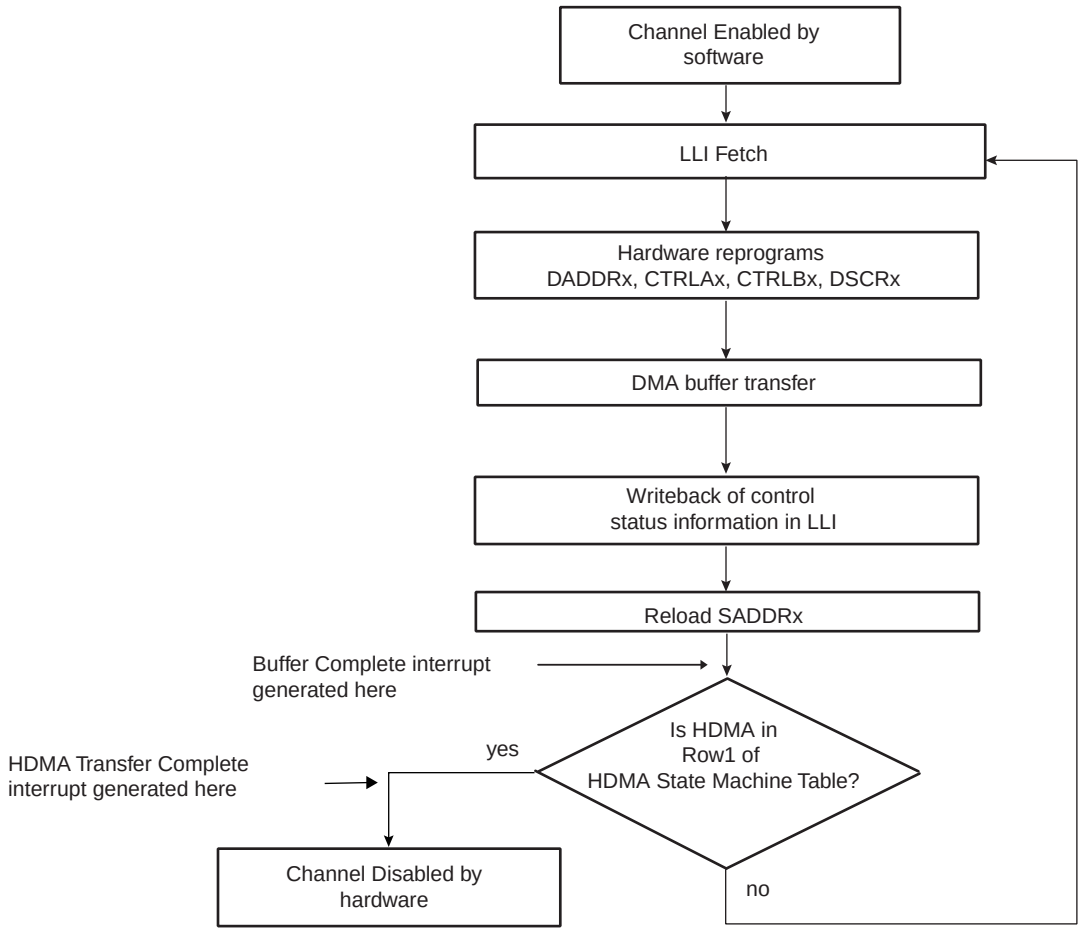


Figure 40-12. DMAC Transfer Flow for Replay Mode at Source and Linked List Destination Address



40.5.5.6 Multi-buffer Transfer with Source Address Auto-reloaded and Contiguous Destination Address (Row 11)

1. Read the Channel Enable register to choose a free (disabled) channel.
2. Clear any pending interrupts on the channel from the previous DMAC transfer by reading to the Interrupt Status Register.
3. Program the following channel registers:
   a. Write the starting source address in the DMAC_SADDRx register for channel x.
   b. Write the starting destination address in the DMAC_DADDRx register for channel x.
   c. Program DMAC_CTRLAx, DMAC_CTRLBx and DMAC_CFGx according to Row 11 as shown in Table 40-2 on page 979. Program the DMAC_DSCRx register with ‘0’. DMAC_CTRLBx.AUTO field is set to ‘1’ to enable automatic mode support.
   d. Write the control information for the DMAC transfer in the DMAC_CTRLBx and DMAC_CTRLAx register for channel x. For example, in this register, you can program the following:
   – i. Set up the transfer type (memory or non-memory peripheral for source and destination) and flow control device by programming the FC of the DMAC_CTRLBx register.
   – ii. Set up the transfer characteristics, such as:
   – Transfer width for the source in the SRC_WIDTH field.
   – Transfer width for the destination in the DST_WIDTH field.
   – Source AHB master interface layer in the SIF field where source resides.
   – Destination AHB master interface master layer in the DIF field where destination resides.

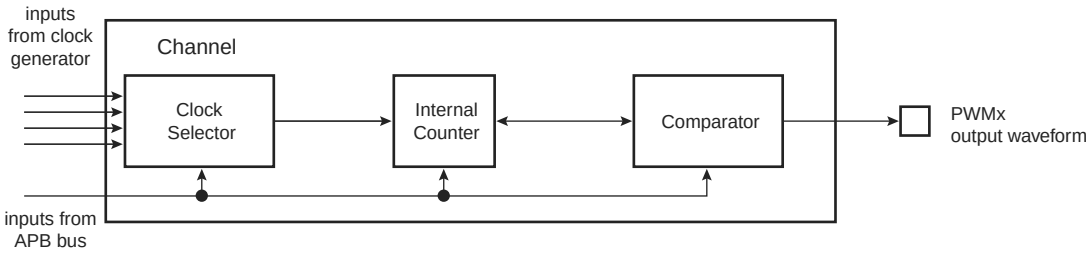
After a reset of the PWM controller, DIVA (DIVB) and PREA (PREB) in the PWM Mode register are set to 0. This implies that after reset clkA (clkB) are turned off.

At reset, all clocks provided by the modulo n counter are turned off except clock “clk”. This situation is also true when the PWM master clock is turned off through the Power Management Controller.

41.6.2 PWM Channel

41.6.2.1 Block Diagram

Figure 41-3. Functional View of the Channel Block Diagram



Each of the 4 channels is composed of three blocks:

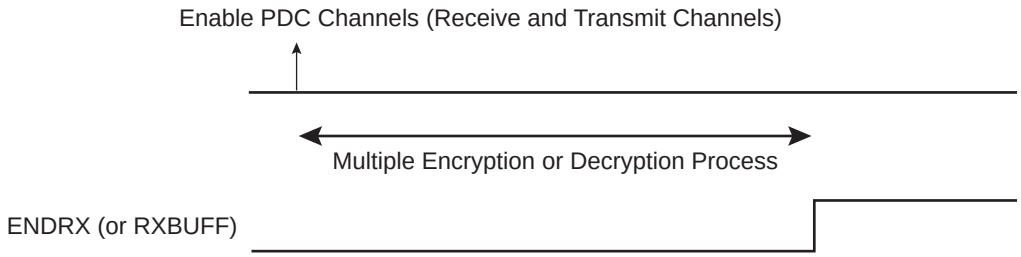
- A clock selector which selects one of the clocks provided by the clock generator described in Section 41.6.1 “PWM Clock Generator” on page 1027.
- An internal counter clocked by the output of the clock selector. This internal counter is incremented or decremented according to the channel configuration and comparators events. The size of the internal counter is 16 bits.
- A comparator used to generate events according to the internal counter value. It also computes the PWMx output waveform according to the configuration.

45.3.3.2 PDC Mode

TDES_MR.LOD = 0

The end of the encryption/decryption is indicated by the ENDRX (or RXBUFF) flag rise in the TDES_ISR. See Figure 45-3.

Figure 45-3. PDC Mode with LOD = 0:



TDES_MR.LOD = 1

The user must first wait for the ENDTX (or TXBUFE) flag to rise in the TDES_ISR, then for DATRDY to ensure that the encryption/decryption is completed. See Figure 45-4.

In this case, no receive buffers are required.

The output data is only available on the Output Data Registers (TDES_ODATARx).

Figure 45-4. PDC Mode with LOD = 1

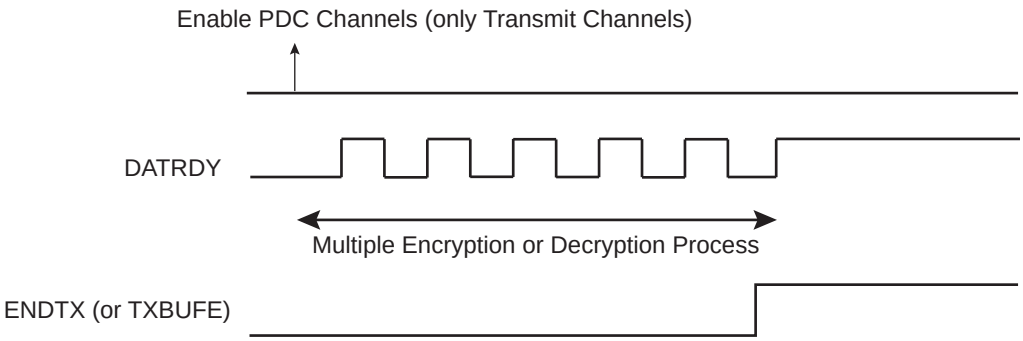


Table 45-4 summarizes the different cases.

Table 45-4. Last Output Mode Behavior versus Start Modes

Sequence	Manual and Auto Modes		PDC Mode	
	LOD = 0	LOD = 1	LOD = 0	LOD = 1
DATRDY Flag Clearing Condition (1)	At least one Output Data Register must be read	At least one Input Data Register must be written	Not used	Managed by the PDC
Encrypted/Decrypted Data Result Location	In the Output Data Registers	In the Output Data Registers	At the address specified in the Receive Pointer Register (TDES_RPR)	In the Output Data Registers
End of Encryption/Decryption	DATRDY	DATRDY	ENDRX (or RXBUFF)	ENDTX (or TXBUFE) then DATRDY

Note: 1. Depending on the mode, there are other ways of clearing the DATRDY flag. See “TDES Interrupt Status Register” on page 1126.

46.5.7 SHA Input Data x Register

Name: SHA_IDATAxR

Access: Write-only

31	30	29	28	27	26	25	24
IDATAx							
23	22	21	20	19	18	17	16
IDATAx							
15	14	13	12	11	10	9	8
IDATAx							
7	6	5	4	3	2	1	0
IDATAx							

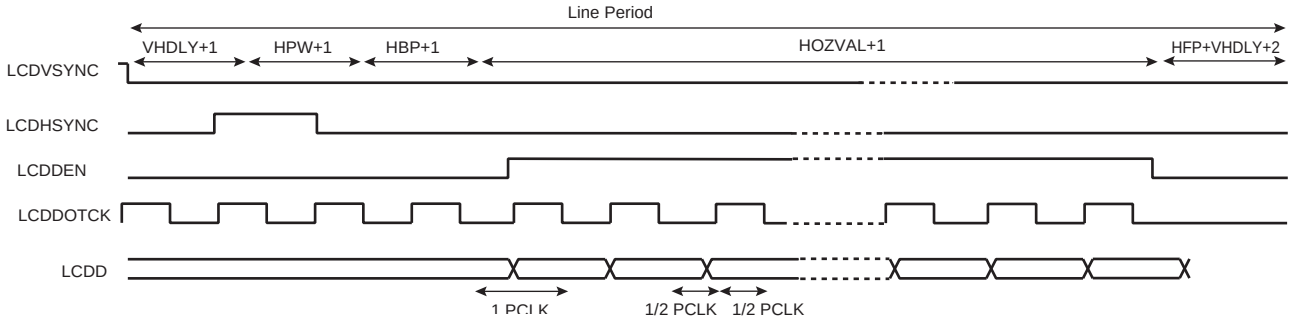
- **IDATAx: Input Data x**

The two 32-bit Input Data registers allow to set the 512-bit data block used for hash processing.

IDATA1 corresponds to the first word of the 512-bit block, and IDATA16 to the last one.

These registers are write-only to prevent the input data from being read by another application.

Figure 47-7. TFT Panel Timing (Line Expanded View), CLKMOD = 1



Usually the LCD_FRM rate is about 70 Hz to 75 Hz. It is given by the following equation:

$$\frac{1}{f_{LCDVSYNC}} = \left(\frac{VHDLY + HPW + HBP + HOZVAL + HFP + (4)}{f_{LCDDOTCK}}\right)(VBP + LINEVAL + VFP + 1)$$

where:

- HOZVAL determines de number of LCDDOTCK cycles per line
- LINEVAL determines the number of LCDHSYNC cycles per frame, according to the expressions shown below:

In STN Mode:

$$HOZVAL = \frac{Horizontal\_display\_size}{Number\_data\_lines} - 1$$

$$LINEVAL = Vertical\_display\_size - 1$$

In monochrome mode, Horizontal_display_size is equal to the number of horizontal pixels. The number_data_lines is equal to the number of bits of the interface in single scan mode; number_data_lines is equal to half the bits of the interface in dual scan mode.

In color mode, Horizontal_display_size equals three times the number of horizontal pixels.

In TFT Mode:

$$HOZVAL = Horizontal\_display\_size - 1$$

$$LINEVAL = Vertical\_display\_size - 1$$

The frame rate equation is used first without considering the clock periods added at the end beginning or at the end of each line to determine, approximately, the LCDDOTCK rate:

$$f_{lcd\_pclk} = (HOZVAL + 5) \times (f_{lcd\_vsync} \times (LINEVAL + 1))$$

With this value, the CLKVAL is fixed, as well as the corresponding LCDDOTCK rate.

Then select VHDLY, HPW and HBP according to the type of LCD used and “Equation 1” on page 1161.

Finally, the frame rate is adjusted to 70–75 Hz with the HFP value:

$$HFP = f_{LCDDOTCK} \times \left[\frac{1}{f_{LCDVSYNC} \times (LINEVAL + VBP + VFP + 1)}\right] - (VHDLY + HPW + HPB + HOZVAL + 4)$$

The line counting is controlled by the read-only field LINECNT of LCDCON1 register. The LINECNT field decreases by one unit at each falling edge of LCDHSYNC.

47.11 Register Configuration Guide

Program the PIO Controller to enable LCD signals.

Enable the LCD controller clock in the Power Management Controller.

47.11.1 STN Mode Example

STN color(R,G,B) 320*240, 8-bit single scan, 70 frames/sec, Master clock = 60 MHz

Data rate: 320*240*70*3/8 = 2.016 MHz

HOZVAL = ((3*320)/8) - 1

LINEVAL = 240 -1

CLKVAL = (60 MHz/2.016 MHz) - 1 = 29

LCDCON1 = CLKVAL << 12

LCDCON2 = LITTLEENDIAN | SINGLESCAN | STNCOLOR | DISP8BIT| PS8BPP;

LCDTIM1 = 0;

LCDTIM2 = 10 | (10 << 21);

LCDFRMCFG = (HOZVAL << 21) | LINEVAL;

LCDMVAL = 0x80000004;

DMAFRMCFG = (7 << 24) + (320 * 240 * 8) / 32;

47.11.2 TFT Mode Example

This example is based on the NEC TFT color LCD module NL6448BC20-08.

TFT 640*480, 16-bit single scan, 60 frames/sec, pixel clock frequency = [21 MHz..29 MHz] with a typical value = 25.175 MHz.

The Master clock must be (n + 1)*pixel clock frequency

HOZVAL = 640 - 1

LINEVAL = 480 - 1

If Master clock is 100 MHz

CLKVAL = (100 MHz / 25.175 MHz) - 1 = 3

VFP = (12 -1), VBP = (31-1), VPW = (2-1), VHDLY = (2-1)

HFP = (16-1), HBP = (48 -1), HPW = (96-1)

LCDCON1 = CLKVAL << 12

LCDCON2 = LITTLEENDIAN | CLKMOD | INVERT_CLK | INVERT_LINE | INVERT_FRM | PS16BPP | SINGLESCAN | TFT

LCDTIM1 = VFP | (VBP << 8) | (VPW << 16) | (VHDLY << 24)

LCDTIM2 = HBP | (HPW << 8) | (HFP << 21)

LCDFRMCFG = (HOZVAL << 21) | LINEVAL

LCDMVAL = 0

DMAFRMCFG = (7 << 24) + (640 * 480* 16) / 32;