TF-530-AC

Wireless Digital TV Streaming Controller

Datasheet

DS_TF-530_001R005
Version: 2.00
12/31/2004

Revision History

Revision	Date	Change Description
DS_TF-530_001R001 V0.01	10/10/2003	Preliminary
DS_TF-530_001R002 V0.90	12/13/2003	Preliminary
DS_TF-530_001R003 V1.00	02/23/2004	Preliminary
DS_TF-530_001R004 V1.02	05/17/2004	Change IO type IU/OD to IU/O Change IO type ID/OD to ID/O
DS_TF-530_001R005 V2.00	11/29/2004	Update for B0 version Modify the DC characteristic

Contents

1. Description

Taifatech's TF-530-AC Wireless Digital TV Streaming Controller provides a very cost effective, high performance SoC solution for consumer electronic and network video camera manufacturers that need to stream digital TV broadcasts or multimedia contents using the industry standard WiFi wireless technology. The controller has a built-in 10/100 Ethernet MAC, Card Bus/PCMCIA/Mini-PCI interface, streaming video host interface, WLAN to LAN Bridging Engine and is bundled with TCP/IP Protocol and Web Server software. The Card Bus or Min-PCI interface allows the use of different peripherals such as WiFi wireless or Compact Flash card to meet the design needs.

The unique architecture of the TF-530-AC allows different design possibilities whether it is wired or wireless. The built-in 10/100Mbps MAC on the TF-530-AC allows easy connection to the LAN, and the Card Bus/PCMCIA/Mini-PCI interface allows connectivity to WLAN using off the shelf 802.11a/b/g WiFi chipsets.

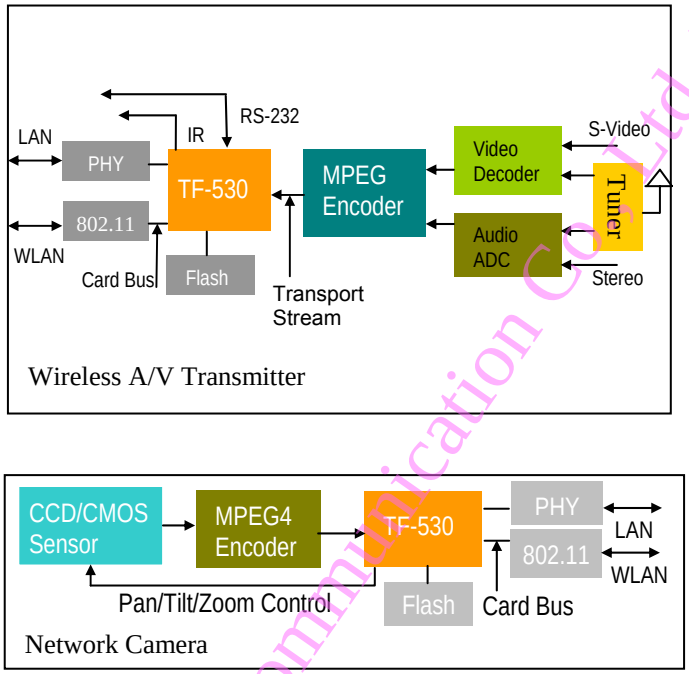
Multimedia contents such as JPEG and MPEG-1,2,4 can also be streamed via the streaming video interface. With the bundled HTTP Server software, the TF-530-AC is the most cost effective solution for implementing systems such as wireless network camera, networked DVD player, Digital Media Adapter, wireless video capture or wireless A/V transport devices.

1.1 Features

- **Streaming Video Interface**
  - Seamless connection to JPEG and MPEG encoders & decoders
  - Hardware support for TS and PS packets
- **PC Card Interface Controller**
  - Dedicated pipelined RISC core for device control, 66MHz operating frequency
  - 32KB program, 4KB data & 20KB packet buffer
  - Selectable PCMCIA, Card Bus or Mini-PCI mode
- **WLAN to LAN Bridging Engine**
  - Hardware 802.11 & 802.3 format translation
  - Hardware Wireless to Ethernet bridging translation
- **Built-in 10/100M Ethernet MAC**
  - Selectable MII or Reverse MII interface
  - 802.3x flow control for full duplex mode and Jamming for half duplex mode
- **Web Server Controller**
  - Pipelined RISC core, 66MHz operating frequency
  - 16KB memory for data, packet buffer & program mirroring (4K, 8K or 12K bytes)
  - Supports up to 2MB External Flash/ROM/SRAM
  - External Flash write support
- **CPU/Video/LAN/WLAN Switch**
  - High speed inter-connectivity
- **4-entry Layer4 address forwarding table for TCP/UDP packet forwarding**
- **256-entry L2 address table for packet filtering & forwarding with port-based direct route support**
- **Two sets of Extended I/O**
- **8-bit customer ID support through bonding option**
- **Supports WDT for H/W and S/W fatal error protection**
- **Internet Protocol Accelerator support**
- **0.18 um CMOS technology, 1.8V internal operation, 3.3V IOs**
- **144-pin LQFP package**

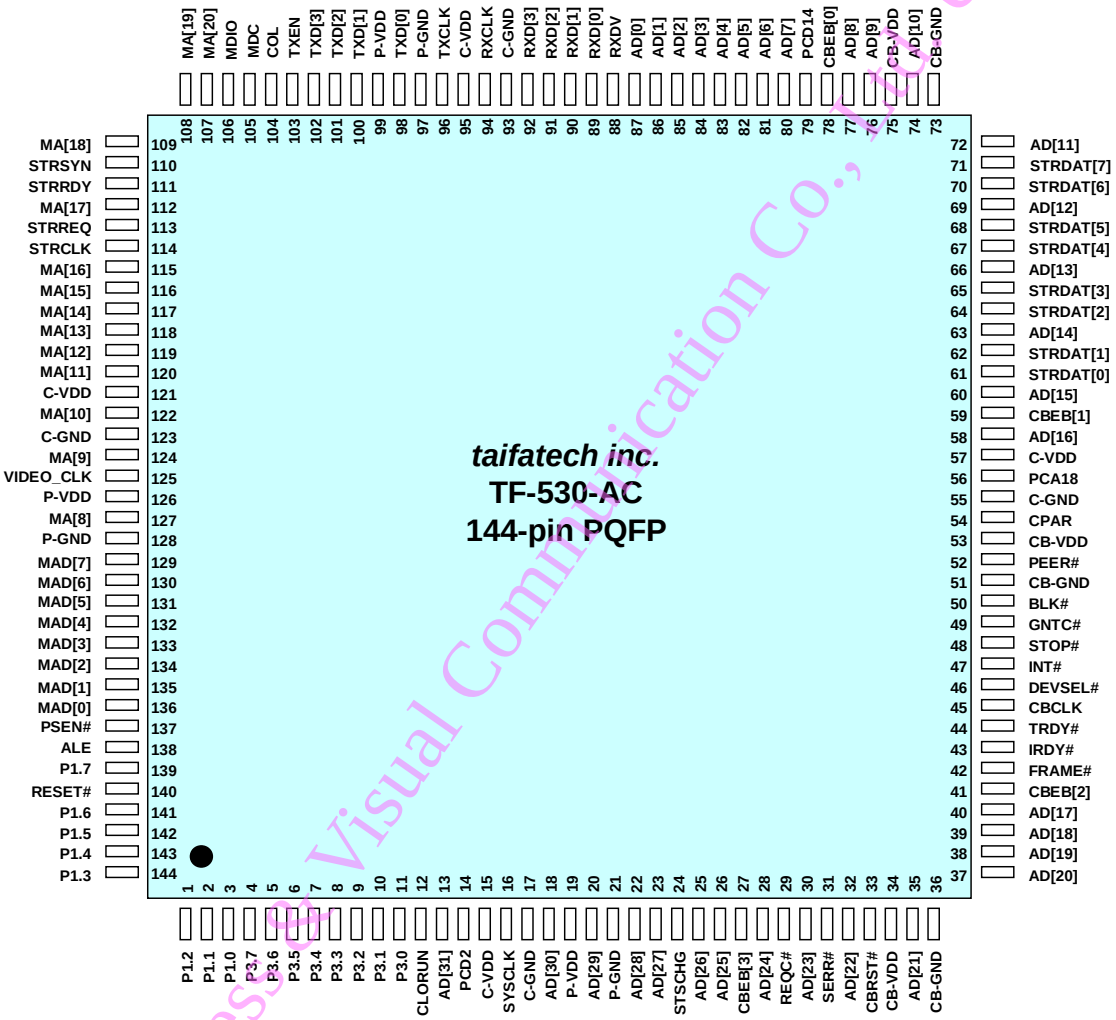
2. System Block Diagram

System block diagrams for TF-530-AC board design are shown below.



3. Pin Description

3.1 Top view of pin out



3.2 PC Card Interface – Card Bus/ PCMCIA

Pin Name	No.	Type	Description
GNTC#/ *PCWE#*	49	O/O	**Cardbus Grant.** This signal indicates that the bus request from another master device has been accepted. **PCMCIA write enable**. This signal is active low and is asserted during memory write transfers to PCMCIA cards.
CBCLK/ *PCA16*	45	O/O	**Cardbus Clock.** The CARDBUS inputs for CARDBUS relative signals synchronization. The frequency range of CCLK is limited from 20MHz to 33MHz. **PCMCIA address pin 16**:
CBRST#/ *PCRST*	33	O/O	**Cardbus Reset.** Active low. CARDBUS signal to initialize slave devices. It remains in high impedance when power on. The active reset signal must be sustained at least 100us to guarantee the slave devices have completed the initialization activity. **PCMCIA reset**: Active high. The signal forces the PC Card to reset internal devices and clears its configuration registers. It remains in high impedance when power on. The active reset signal must be sustained for 10us as specified by the PCMCIA standard.
PCA18	56	O/O	**PCMCIA address pin 18**
FRAME#/ *PCA23*	42	I/O/O	**Cardbus Cycle Frame.** This signal is driven by master device to occupy the bus for data transfer. This signal is driven at the beginning and during bus access. When FRAME# is kept active, data transfer continues. When FRAME# is deasserted, the transaction is in the final data phase. As a target device, the device monitors this signal before decoding the address to check if the current transaction is addressed to it. **PCMCIA address pin 23**
PCD14	79	I/O/I/O	**PCMCIA data pin 14**
PCD2	14	I/O/I/O	**PCMCIA data pin 2**
IRDY#/ *PCA15*	43	I/O/O	**Cardbus Initiator Ready.** Master device is ready for data transaction. **PCMCIA address pin 15**
TRDY#/ *PCA22*	44	I/O/O	**Cardbus Target Ready.** Target device is ready for data transaction. **PCMCIA address pin 22**
CPAR/ *PCA13*	54	I/O/O	**Cardbus Parity**. This signal indicates even parity across AD [31:0] and C_BEB [3:0] including the PAR pin. As a master device, PAR is asserted during address and write data phase. As a target device, PAR is asserted during read data phase. **PCMCIA address pin 13**
PERR#/ *PCA14*	52	I/O/O	**Cardbus Parity Error.** When TF-530-AC is the bus target and detects a data parity error, TF-530-AC asserts this pin low. **PCMCIA address pin 14**
DEVSEL#/ *PCA21*	46	I/O/O	**Cardbus Device select.** As a bus master, this pin indicates the start and duration that a target recognizes the destination address for the data transfer. As a target device, the TF-530-AC asserts this signal low when it recognizes its target address after FRAME# is asserted. **PCMCIA address pin 21**
CBEB[3]/ *REG#*	27	I/O/O	**Cardbus bus command and byte enable 3** **PCMCIA register select**. Asserted this signal specifies access to attribute memory.
CBEB[2]/ *PCA12*	41	I/O/O	**Cardbus bus command and byte enable 2**

			PCMCIA address pin 12
CBEB[1]/ *PCA8*	59	I/O/O	**Cardbus bus command and byte enable 1** **PCMCIA address pin 8**
CBEB[0]/ *CE1#*	78	I/O/O	**Cardbus bus command and byte enable 0** **PCMCIA Card enable 1:** active low. When this signal is asserted, it indicates only the address locations of lower byte data (D7:D0) are valid.
AD[31]/ *PCD10*	13	I/O/I/O	**Cardbus address/data pin 31** **PCMCIA address pin 10**
AD[30]/ *PCD9*	18	I/O/I/O	**Cardbus address/data pin 30** **PCMCIA address pin 9**
AD[29]/ *PCD1*	20	I/O/I/O	**Cardbus address/data pin 29** **PCMCIA data pin 1**
AD[28]/ *PCD8*	22	I/O/I/O	**Cardbus address/data pin 28** **PCMCIA data pin 8**
AD [27]/ *PCD0*	23	I/O/I/O	**Cardbus address/data pin 27** **PCMCIA data pin 0**
AD[26]/ *PCA0*	25	I/O/O	**Cardbus address/data pin 26** **PCMCIA address pin 0**
AD[25]/ *PCA1*	26	I/O/O	**Cardbus address/data pin 25** **PCMCIA address pin 1**
AD[24]/ *PCA2*	28	I/O/O	**Cardbus address/data pin 24** **PCMCIA address pin 2**
AD[23]/ *PCA3*	30	I/O/O	**Cardbus address/data pin 23** **PCMCIA address pin 3**
AD[22]/ *PCA4*	32	I/O/O	**Cardbus address/data pin 22** **PCMCIA address pin 4**
AD[21]/ *PCA5*	35	I/O/O	**Cardbus address/data pin 21** **PCMCIA address pin 5**
AD[20]/ *PCA6*	37	I/O/O	**Cardbus address/data pin 20** **PCMCIA address pin 6**
AD[19]/ *PCA25*	38	I/O/O	**Cardbus address/data pin 19** **PCMCIA address pin 25**
AD[18]/ *PCA7*	39	I/O/O	**Cardbus address/data pin 18** **PCMCIA address pin 7**
AD[17]/ *PCA24*	40	I/O/O	**Cardbus address/data pin 17** **PCMCIA address pin 24**
AD[16]/ *PCA17*	58	I/O/O	**Cardbus address/data pin 16** **PCMCIA address pin 17**
AD[15]/ *PCIOWR#*	60	I/O/O	**Cardbus address/data pin 15** **PCMCIA IO Write command**
AD[14]/ *PCA9*	63	I/O/O	**Cardbus address/data pin 14** **PCMCIA address pin 9**
AD[13]/ *PCIORD#*	66	I/O/O	**Cardbus address/data pin 13** **PCMCIA IO Read command**
AD[12]/ *PCA11*	69	I/O/O	**Cardbus address/data pin 12** **PCMCIA address pin 11**
AD[11]/ *PCOE#*	72	I/O/O	**Cardbus address/data pin 11** **PCMCIA Output enable**
AD[10]/ *PCCE2#*	74	I/O/O	**Cardbus address/data pin 10** **PCMCIA Card enable 2**

AD[9]/ *PCA10*	76	I/O/O	**Cardbus address/data pin 9** **PCMCIA address pin 10**
AD[8]/ *PCD15*	77	I/O/I/O	**Cardbus address/data pin 8** **PCMCIA address pin 15**
AD[7]/ *PCD7*	80	I/O/I/O	**Cardbus address/data pin 7** **PCMCIA data pin 7**
AD[6]/ *PCD13*	81	I/O/I/O	**Cardbus address/data pin 6** **PCMCIA data pin 13**
AD[5]/ *PCD6*	82	I/O/I/O	**Cardbus address/data pin 5** **PCMCIA data pin 6**
AD[4]/ *PCD12*	83	I/O/I/O	**Cardbus address/data pin 4** **PCMCIA data pin 12**
AD[3]/ *PCD5*	84	I/O/I/O	**Cardbus address/data pin 3** **PCMCIA address pin 9**
AD[2]/ *PCD11*	85	I/O/I/O	**Cardbus address/data pin 2** **PCMCIA data pin 11**
AD[1]/ *PCD4*	86	I/O/I/O	**Cardbus address/data pin 1** **PCMCIA data pin 4**
AD[0]/ *PCD3*	87	I/O/I/O	**Cardbus address/data pin 0** **PCMCIA data pin 3**
STOP#/ *PCA20*	48	I/O/O	**Cardbus Stop.** Target device requests the master device to stop the current transaction. **PCMCIA address pin 20**
CLKRUN#/ *IOIS16#*	12	I/O/O	**Cardbus Clock Run**. This signal controls the clock run/stop protocol. **PCMCIA IO Size is 16-bits.**
BLK#/ *PCA19*	50	I/O/O	**Cardbus Block:** No use in Cardbus mode(TF-530-AC does not support Cardbus block mode) **PCMCIA address pin 19**
REQC#/ *INPACK#*	29	I/I	**Cardbus Request.** This signal indicates that another bus master wants to get the bus access right. **PCMCIA Input port acknowledge**: no use in our chip.
SERR#/ *WAIT#*	31	I/I	**Cardbus System Error.** If an address parity error is detected, TF-530-AC asserts this pin low. **PCMCIA Wait signal**
INT#/ *IREQ*	47	I/I	**Cardbus INT.** CARDBUS interrupt request. **PCMCIA Interrupt request**
CSTSCHG/ *STSCHG#*	24	I/I	**Cardbus Status change**. Active high. The notified signal for Card status change, such as “wake up”, “Ready/Busy”, “Write protect”, “Battery voltage detect”. **PCMCIA Status change**. Active low.

3.3 Memory/Peripheral Interface

Pin Name	No.	Type	Description
P3.7/RD# **P3.6/WR#** **P3.5/T1/TDI** **P3.4/T0/TDO** **P3.3/INT1#** **P3.2/XFWR#** **P3.1/STXD1** **P3.0/SRXD1**	4 5 6 7 8 9 10 11	IU/O	Port 3. 8-bit bi-directional I/O port. Alternate function: **RD#** External Data Memory read strobe output **WR#** External Data Memory write strobe output **T1** Timer 1 external input **TDI** Debugger Interface data input **T0** Timer 0 external input **TDO** Debugger Interface data output **INT1#** External Interrupt input 1 **XFWR#** External flash write strobe output **STXD1** Serial Port 1 Transmit output **SRXD1** Serial Port 1 Receive input
P1.7 **P1.6** **P1.5** **P1.4/ IODMACS#/ IR** **P1.3/ SRAMCS#** **P1.2/ IORDY#** **P1.1/ IOCS1#/ TCK** **P1.0/ IOCS0#**	139 141 142 143 144 1 2 3	IU/O	Port 1. 8-bit bi-directional I/O port. Alternate function: **TCK** Debugger Interface clock input **IR** Remote Control Infra-Red input/output **SRAMCS#** SRAM chip select output **IORDY#** Extended IO ready input **IOCS1#** Extended IO chip select 1 output **IOCS0#** Extended IO chip select 0 output
PSEN#	137	O	**Program Store Enable Output**. This signal is commonly connected to external program memory as a chip enable. PSEN# provides an active low pulse and is driven high when external program memory is not being accessed.
ALE	138	O	**Address Latch Enable Output**. This signal is commonly connected to latch external memory address. ALE provides an active high pulse to latch memory address
MAD [7:0]	129,130, 131,132, 133,134, 135,136	I/O	Address/Data bus.
MA[15:8]	116, 117,118, 119,120, 122,124, 127	I/O	Address Bus 15 ~ 8
MA[16]	115	IU/O	Address Bus 16 Pull low with 1KΩ resistor for normal operation
MA[17]	112	ID/O	Address bus 17 Pull low with 1KΩ resistor for normal operation
MA[18]	109	IU/O	Address bus 18 Pull up with 4.7KΩ resistor for normal operation
MA[19]/ SRXD2/P4.0	108	IU/O	Address bus 19 Alternate function: **SRXD2** Serial Port 2 Receive input

			P4.0 bi-directional I/O port.
MA[20]/ STXD2/P4.1	107	IU/O	Address bus 20 Alternate function: **STXD2** Serial Port 2 Transmit output **P4.1** bi-directional I/O port.

3.4 MII/ RvMII Interface

Pin Name	No.	Type	Description
TXD [3:0]/ RvRXD [3:0]	102,101,100,98	O	**MII Transmit Data.** TF-530-AC sources TXD [3:0] synchronous with TXCLK when TXEN is asserted. TXD[3:1] are also used for power on setting. **Reverse MII Receive Data**. TF-530-AC sources RvRXD [3:0] synchronous with **RvRXCLK** when **RvRXDV** is asserted.
TXCLK/ RvRXCLK	96	I	**MII Transmit Clock.** Continuous (25MHz/2.5MHz) clock input used to synchronize TXEN and TXD [3:0]. **Reverse MII Receive Clock**. Continuous (25MHz/2.5MHz) clock output used to synchronize **RvRXDV** and RvRXD [3:0].
TXEN/ RvRXDV	103	O	**MII Transmit Enable.** Indicates TF-530-AC has valid data on the TXD [3:0]. **Reverse MII Receive Data Valid**. Indicates TF-530-AC has valid data on the RvRXD [3:0].
RXD [3:0]/ RvTXD [3:0]	92,91,90,89	I	**MII Receive Data.** External device (PHY) will source RXD [3:0] synchronous with RXCLK when RXDV is asserted. **Reverse MII Transmit Data**. External device (MAC) will source RvTXD [3:0] synchronous with **RvTXCLK** when **RvTXEN** is asserted.
RXCLK/ RvTXCLK	94	I	**MII Receive Clock (input).** Continuous (25MHz/2.5MHz) clock input used by MAC to synchronize RXDV, RXD [3:0] and RXER. **Reverse MII Transmit Clock (output)**.
RXDV/ RvTXEN	88	I	**MII Receive Data Valid (input).** RXDV is asserted. It means the external PHY has valid recovered data on the RXD [3:0]. **Reverse MII Transmit Enable (input)**.
COL/ RvCOL	104	I/O	**MII Collision Detection (input).** Active when collision is detected. **Reverse MII Collision Detection (output)**.
MDC	105	O	**Management Data Clock.**
MDIO	106	I/O	**Management Data I/O.**

3.5 Streaming Video Interface

The Video interface pins mapping are controlled by video interface mode and interface direction of Video Port Control register (0x71C0). When video port enable bit of Video Port Control register is not enabled, video port interface except VIDEO_CLK will be IU pad type.

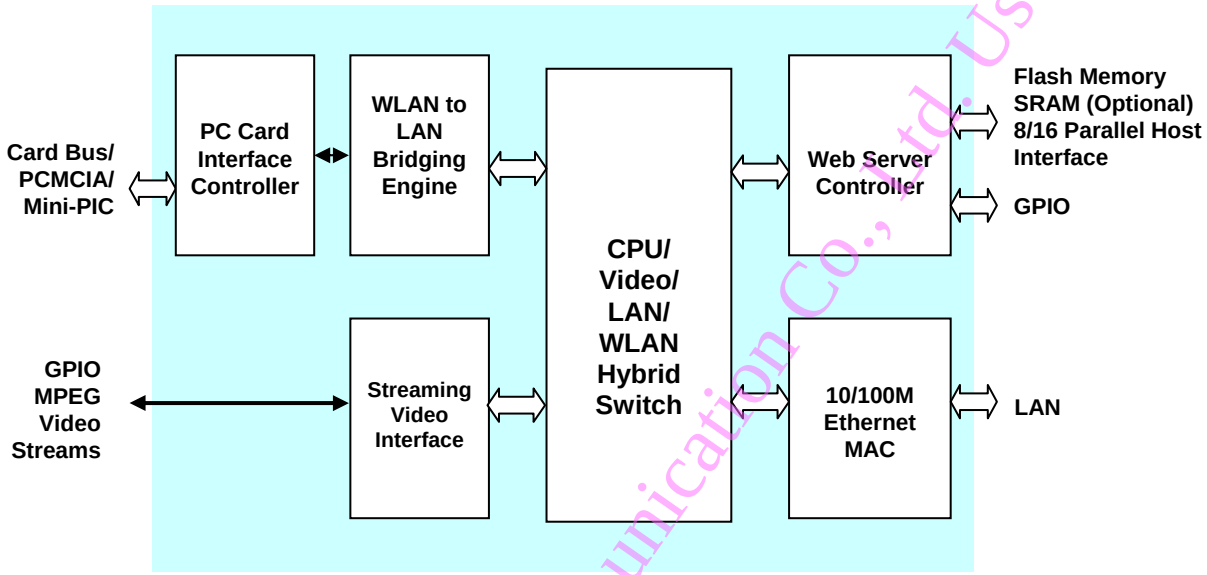
Pin Name	No.	Type	Description
STRSYN	110	IU/O	Streaming Synchronization. Indicate first byte of Packet.
STRCLK	114	IU/O	Streaming Clock.
STRRDY	111	IU/O	Streaming Ready. Indicate Ready to transmit.
STRREQ	113	IU/O	Streaming Request. Request data transmit in.
STRDAT[7:0]	71,70,68,67, 65,64,62,61	IU/O	Streaming Data.
VIDEO_CLK	125	I	27Mhz Video clock input

3.6 Miscellaneous

Pin Name	No.	Type	Description
RESET#	140	IS	**System Reset input**. Active low
SYSCLK	16	I	**System clock input.**
P-VDD	19,99,126	3.3V IO power	
C-VDD	15,57,95,121	1.8V Core power	
CB-VDD	34,53,75	Card Bus 3.3V IO power	
P-GND	21,97,128	IO Ground	
C-GND	17,55,93,123	Core Ground	
CB-GND	36,51,73	Card Bus Ground	
Note: # =active low signal; ID=input with internal pull-down; IU=input with internal pull-up; OD=open drain output; IS=input with Schmitt Trigger			

4. Functional Description

The figure below shows the internal block diagram for TF-530-AC.



TF-530-AC device architecture includes Web Server Controller (WSC), 10/100M Ethernet MAC, CPU/Video/LAN/WLAN Hybrid Switch, WLAN to LAN Bridging Engine, PC Card Interface Controller and Streaming Video Interface.

4.1 Web Server Controller

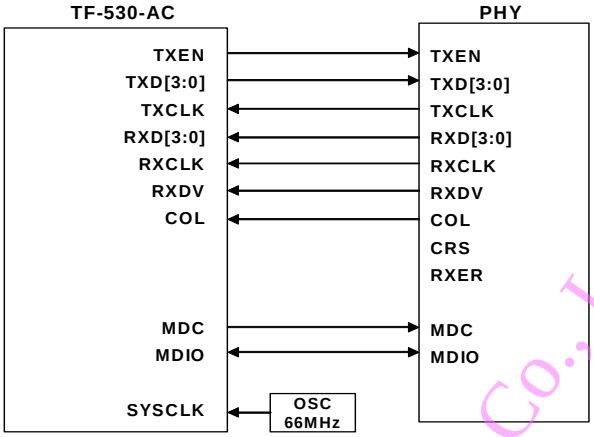
The WSC has an 8-bit pipelined RISC core combined with 16KB internal SRAM. The 16KB SRAM is used for storing the received packets, transmitted packets, and data variables, and for mirroring external flash program stored in external flash. The external flash and external SRAM space can be up to 2MB individually, and the space is from 0x000000 to 0x1FFFFF. However, the on-chip SRAM occupies 128KB memory space, therefore the external memory space is allocated from 0x020000 to 0x1FFFFF.

TF-530-AC provides an 8-bit and 16-bit host interfaces to access external devices connected to Memory/Peripheral bus. Both 8-bit and 16-bit modes work on the same bus. The16-bit interface supports Big-Endian format and Little-Endian formats, and can be applied to external Extended IO #0 only. In order to connect with the interface of various devices, extended IO supports ready-control access and wait-state control access methods. In ready-control access method, TF-530-AC issues a read/write operation to access external device. The access transaction will be continued until external device responds an acknowledgement. However, in wait-state control access method, TF-530-AC issues a read/write operation, and waits a periods, then terminates this access transaction by itself. The space of extended IO is adjustable and is the multiple of 64KB. The default space is 64KB for each extended IO. For extended IO #0, the default space is from 0x1E0000 to 0x1EFFFF, and for extended IO #1 from 0x1F0000 to 0x1FFFFF.

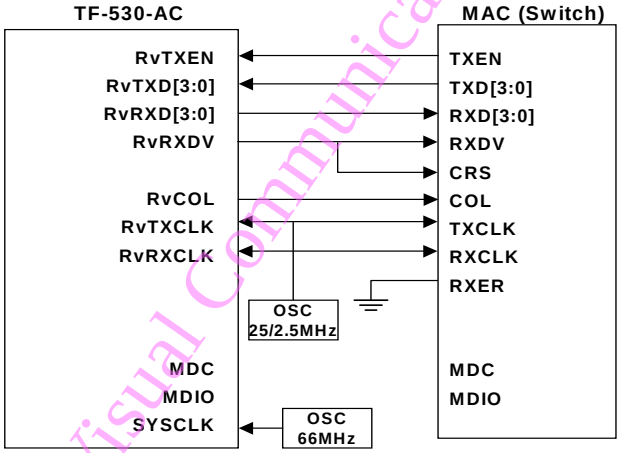
4.2 Network Interface

TF-530-AC employs a 10/100Mbps MAC with MII/ Reverse MII interface for connection with the Physical layer (PHY) and MII serial management interface to read/write registers in PHY. TF-530-AC's MAC supports IEEE 802.3x flow control for full

duplex mode, and jamming flow control mechanism for half duplex mode. The following figures show the connection between TF-530-AC with PHY and MAC (or switch).



Connection with PHY



Connection with MAC

4.3 CPU/Video/LAN/WLAN Hybrid Switch

A 4-port high speed hybrid switch engine is embedded in TF-530-AC for packet switching. The switch engine adopts shared memory architecture. It embeds a 32KB SRAM for packet buffer. The buffer is formatted into 512B blocks. For efficiency of the buffer utilization, user can allocate proper number of blocks for each switch port based on application. Packet drop threshold registers are also used for control. The packet drop threshold registers limit the total number of blocks for each switch ports. When the total used block of one of the switch ports reaches the threshold, incoming packets are dropped until the used block number is under the threshold.

The switch engine also makes the decision of the packets' destination port when packets from one port enter the switch fabric. The algorithm is based on the Destination MAC and IP addresses of the packet and the Layer2 filtering table which records the

information of stations connected to LAN and WLAN. If packet's DA and IP match with one of the entries of Layer2 filtering table, the packet is forwarded to the destination according to the Forward Port Map of this entry.

4.4 WLAN to LAN Bridging Engine

WLAN to LAN Bridging Engine translates the packets from 802.11 format to 802.3 format and from 802.3 format to 802.11 format during DMA data moving from PC Card buffer to the Hybrid Switch (vice versa). The packet translation is fully transparent for 802.11 control and long frames. Packets pass through switch engine should be formatted in 802.3 packet format with one byte frame tag header. The frame tag specifies the frame type and the source port number of packet coming from. While frame type is 802.11 control or long frame, the packet will be passed through switch engine without format translation. The format transfer function can be disabled if necessary.

4.5 PC Card Interface Controller

PC Card Interface Controller supports three interface modes, PCMCIA, Card Bus or Mini-PCI. A dedicated processor is used to control internal resources of PC Card Interface Controller and accesses external device (card). In initial process, Web Server Controller (WSC) handles all resources of PC Card Interface Controller. After WSC downloads program into program memory of the processor and turns it on, then the processor takes over all controls. Of course, WSC can execute all jobs if the processor is always forced in idle state.

4.6 Streaming Video Interface

The Streaming Video Interface (SVI) is designed for connecting to external JPEG and MPEG encoders & decoders to support the most cost effective video streaming applications, such as wireless network camera, networked DVD player wireless DVB tuner or wireless A/V transport devices. To interface numerous venders' devices, the Streaming Video Interface (SVI) supports lots of interface modes, including Transport Stream (TS) and Program Stream (PS) modes. When streaming video data is sent on LAN or WLAN, SVI packs them in 802.3 or 802.11 format and passes the formatted data into the switch. Conversely, when packets from LAN or WLAN need to pass to external JPEG or MPEG decoders, the packet header is stripped. The hooked header content and stripped header length are programmable. The maximum length for the hooked or stripped header is 64 bytes, and the first byte is tag data which is used to indicate the forwarding destination port.

5. Electric Characteristics

5.1 Absolute Maximum Ratings

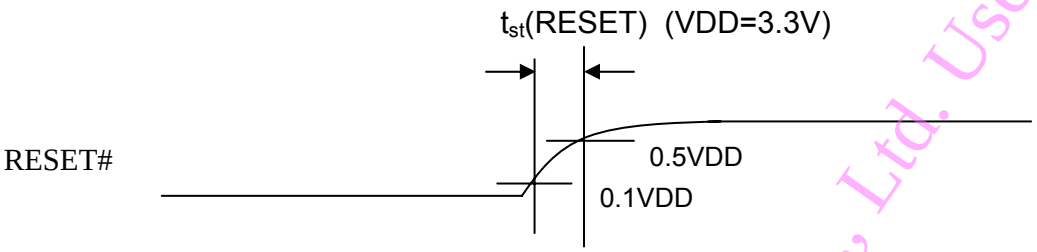
Item	Symbol	Conditions	Rating	Unit
Ambient Temperature	T_{AB}		0 ℃ to 70 ℃	℃
Storage Temperature	T_{STG}		-60 ℃ to 150 ℃	℃
Input Voltage	V_I		-0.3 to V_{DD}+0.5	V

5.2 DC Characteristics (Ta=0℃ to +70℃)

Item	Symbol	Conditions	Min.	Typ.	Max.	Unit
3.3V Power Supply	P-VDD		3.0	-	3.6	V
1.8V Power Supply	C-VDD		1.62	-	1.98	V
Input Low Voltage	V_{IL}		-	-	0.4	V
Input High Voltage	V_{IH}		2.4	-	-	V
Output Low Voltage	V_{OL}	I_{OL}= 2mA	-	-	0.4	V
Output High Voltage	V_{OH}	I_{OH}=-1mA	2.4	-	-	V
Input Leakage Current	I_{IL}	$V_{DD=Max}$, V_I = 0V to 3.3V	-10	-	10	uA
Input Pin Capacitance	C_I	f=1MHz, V_I=0V	-	-	10	pF
Output Pin Capacitance	C_O	f=1MHz, V_I=0V	-	-	10	pF
Input/Output Pin Capacitance	C_{IO}	f=1MHz, V_I=0V	-	-	10	pF
3.3V Active Current	I_{DD33}	f=66MHz (off load)	-	-	50	mA
1.8V Active Current	I_{DD18}	f=66MHz	-	-	180	mA

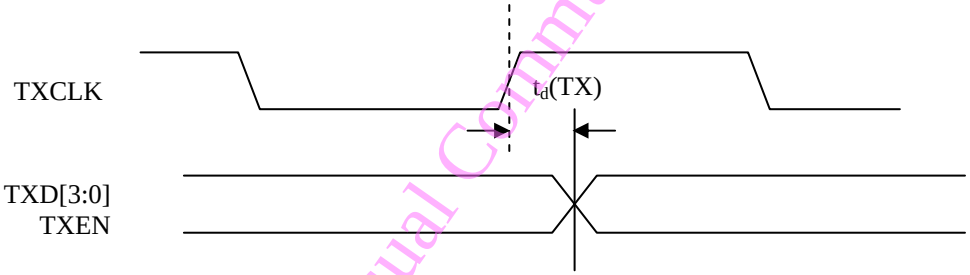
5.3 AC Characteristics Timing Diagram

5.3.1 Reset Timing



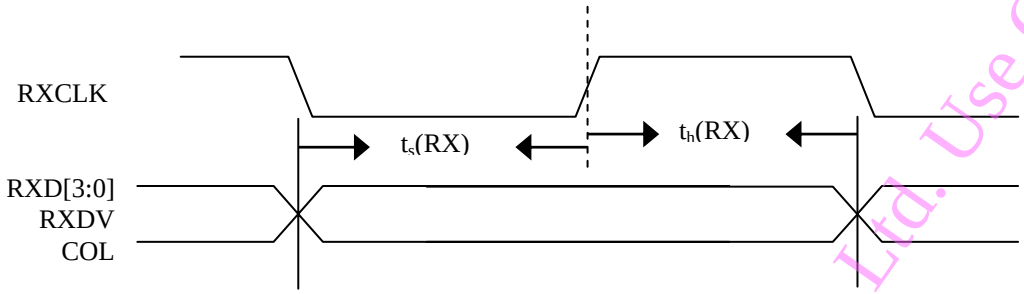
Item	Symbol	Min.	Max.	Unit
RESET time	t_{st}(RESET)	1	-	us

5.3.2 MII TX Timing



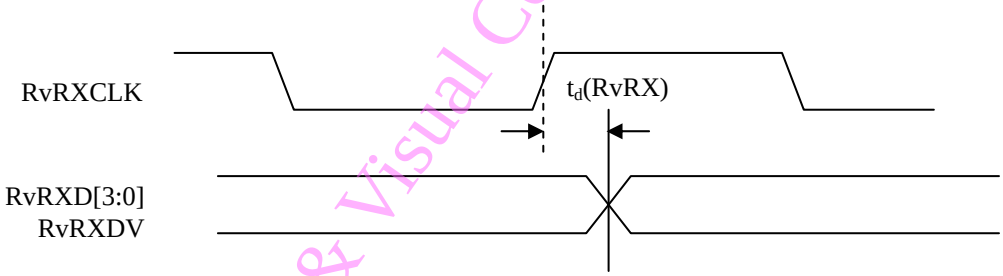
Item	Symbol	Min.	Max.	Unit
MII Output data delay time	t_d(TX)	-	20	ns

5.3.3 MII RX Timing



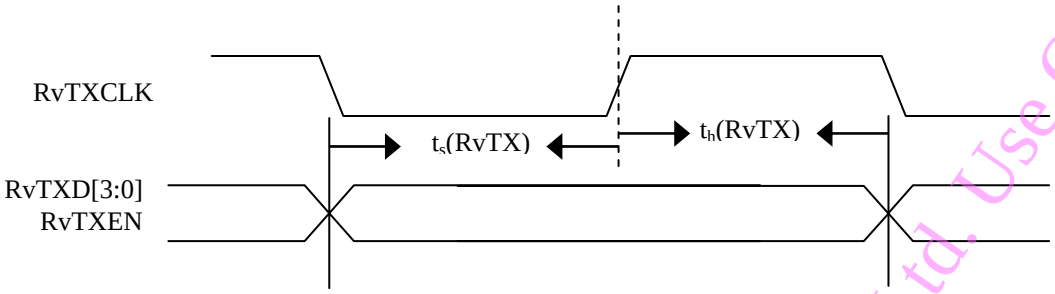
Item	Symbol	Min.	Max.	Unit
MII Input data setup time	$t_{S(RX)}$	5	-	ns
MII Input data hold time	$t_{h(RX)}$	2	-	ns

5.3.4 Reverse MII Output Timing



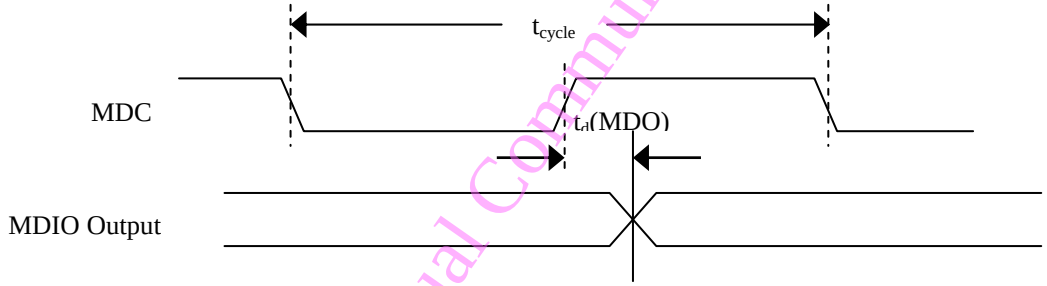
Item	Symbol	Min.	Max.	Unit
RvMII Output data delay time	$t_{d(RvRX)}$	-	20	ns

5.3.5 Reverse MII Input Timing



Item	Symbol	Min.	Max.	Unit
RvMII Input data setup time	$t_{S(RvTX)}$	10	-	ns
RvMII Input data hold time	$t_{h(RvTX)}$	10	-	ns

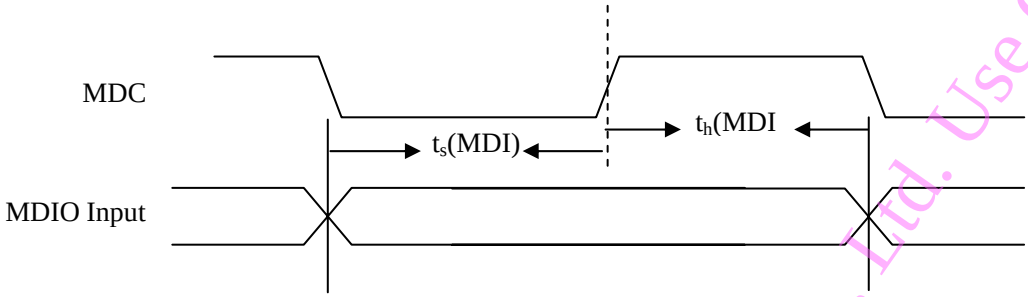
5.3.6 MII Serial Management Output Timing



Item	Symbol	Min.	Max.	Unit
MDIO output data delay time	$t_{d(MDO)}$	-	$(t_{cycle}/2)+10$	ns

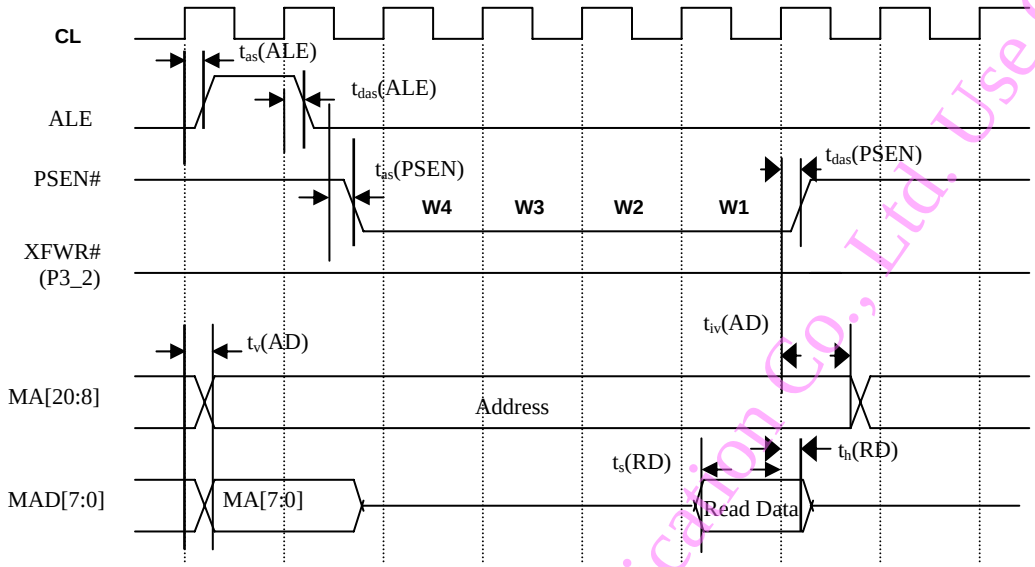
Note: t_{cycle} is the cycle time of the MDC clock, which is configurable

5.3.7 MII Serial Management Input Timing



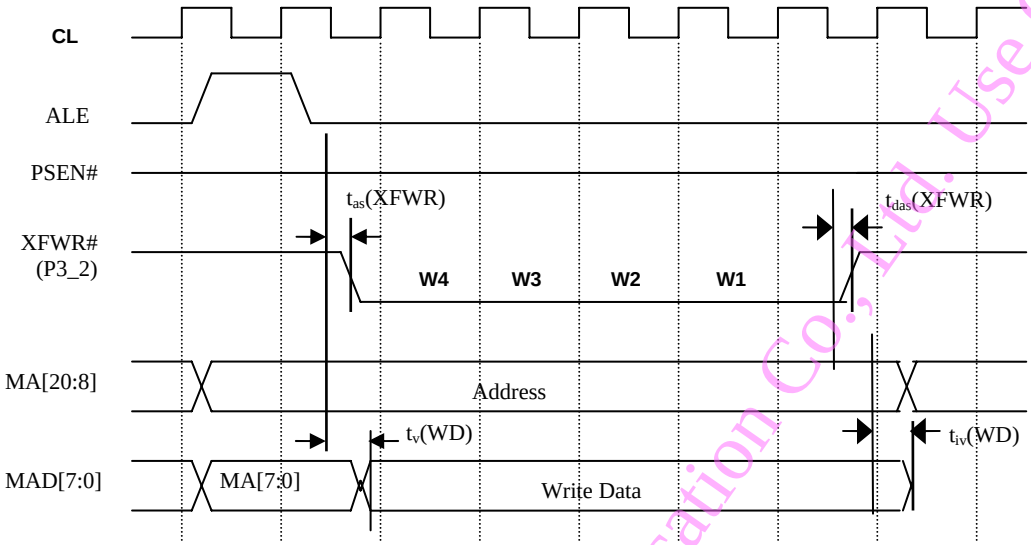
Item	Symbol	Min.	Max.	Unit
MDIO Input data setup time	t_S(MDI)	2	-	ns
MDIO Input data hold time	t_h(MDI)	1	-	ns

5.3.8 External Flash Read Timing with 4 Wait States



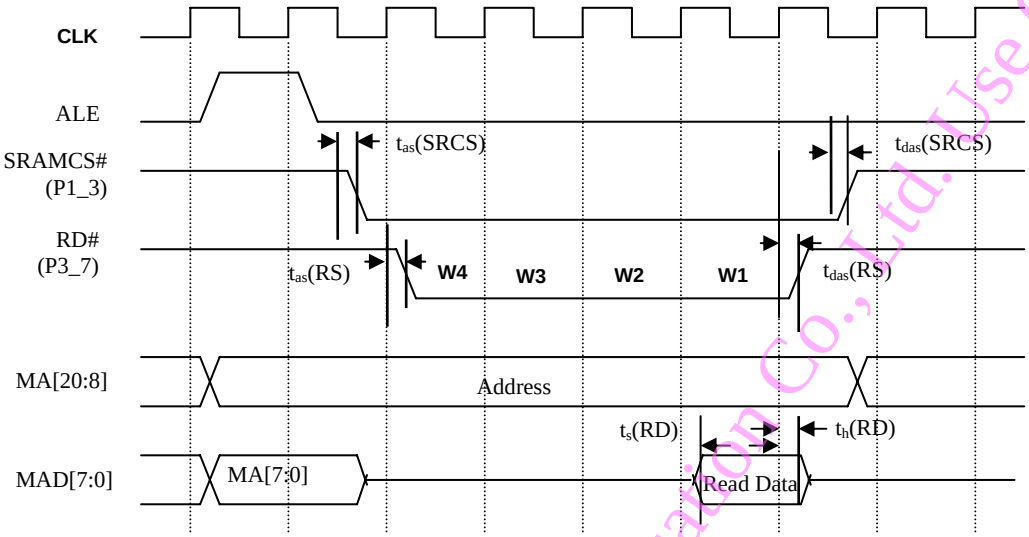
Item	Symbol	Min.	Max.	Unit
Address Latch assert time	$t_{as(ALE)}$	-	12	ns
Address Latch de-assert time	$t_{das(ALE)}$	-	12	ns
Read Strobe assert time	$t_{as(PSEN)}$	-	13	ns
Read Strobe de-assert time	$t_{das(PSEN)}$	-	13	ns
Address/Data valid time	$t_{v(AD)}$	-	12	ns
Address/Data invalid time	$t_{iv(AD)}$	-	15	ns
Read Data Input data setup time	$t_{s(RD)}$	5	-	ns
Read Data Input data hold time	$t_{h(RD)}$	2	-	ns

5.3.9 External Flash Write Timing with 4 Wait States



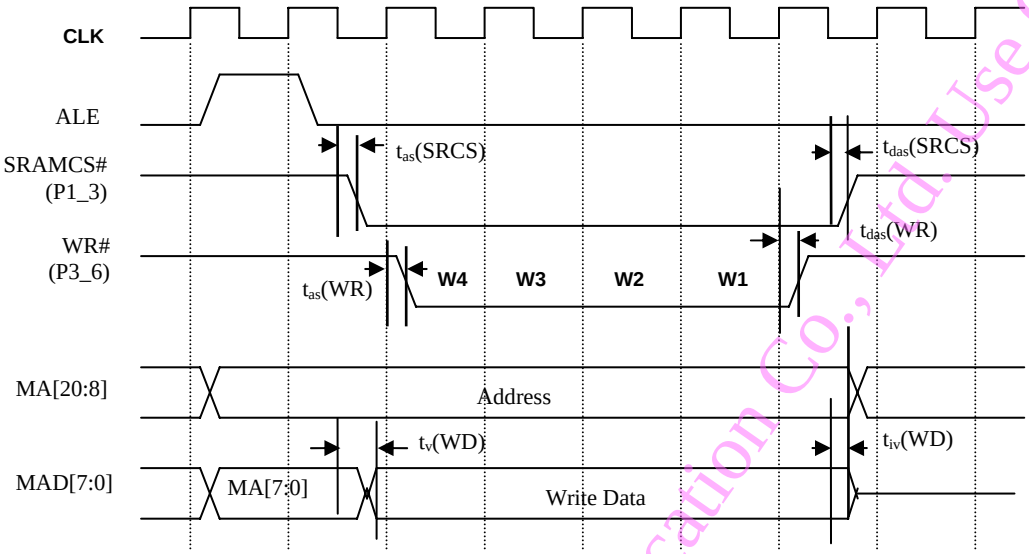
Item	Symbol	Min.	Max.	Unit
Xflash Write assert time	$t_{as(XFWR)}$	-	12	ns
Xflash Write de-assert time	$t_{das(XFWR)}$	-	12	ns
Write Data valid time	$t_{v(WD)}$	-	12	ns
Write Data invalid time	$t_{iv(WD)}$	-	12	ns

5.3.10 External SRAM Read with 4 Wait States



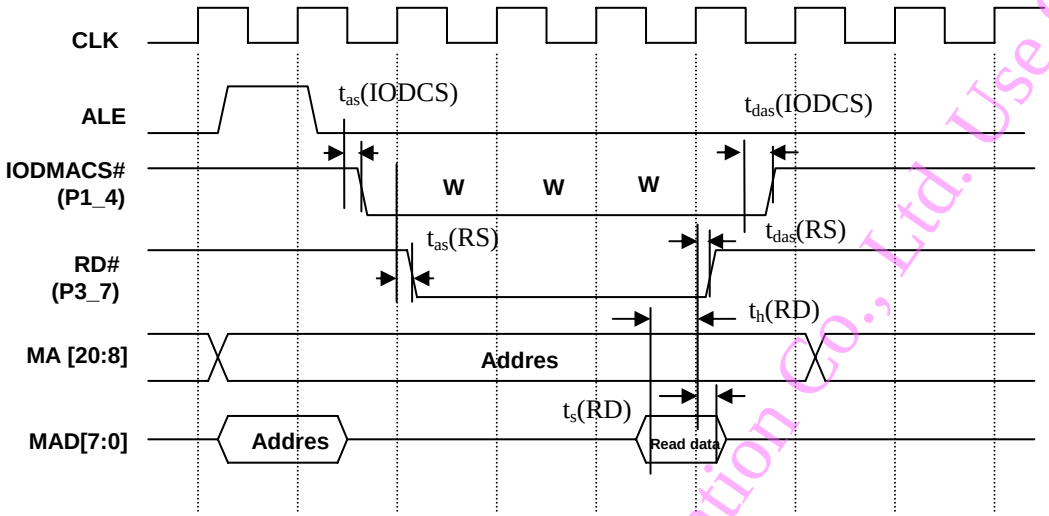
Item	Symbol	Min.	Max.	Unit
XSRAM Chip Select assert time	t_{as}(SRCS)	-	12	ns
XSRAM Chip Select de-assert time	t_{das}(SRCS)	-	12	ns
XSRAM Read Strobe assert time	t_{as}(RS)	-	12	ns
XSRAM Read Strobe de-assert time	t_{das}(RS)	-	12	ns
Read Data Input data setup time	t_{s}(RD)	5	-	ns
Read Data Input data hold time	t_{h}(RD)	2	-	ns

5.3.11 External SRAM Write with 4 Wait States



Item	Symbol	Min.	Max.	Unit
XSRAM Chip Select assert time	t_{as}(SRCS)	-	12	ns
XSRAM Chip Select de-assert time	t_{das}(SRCS)	-	12	ns
XSRAM Write Strobe assert time	t_{as}(WR)	-	12	ns
XSRAM Write Strobe de-assert time	t_{das}(WR)	-	12	ns
Write Data valid time	t_{v}(WD)	-	12	ns
Write Data invalid time	t_{iv}(WD)	-	12	ns

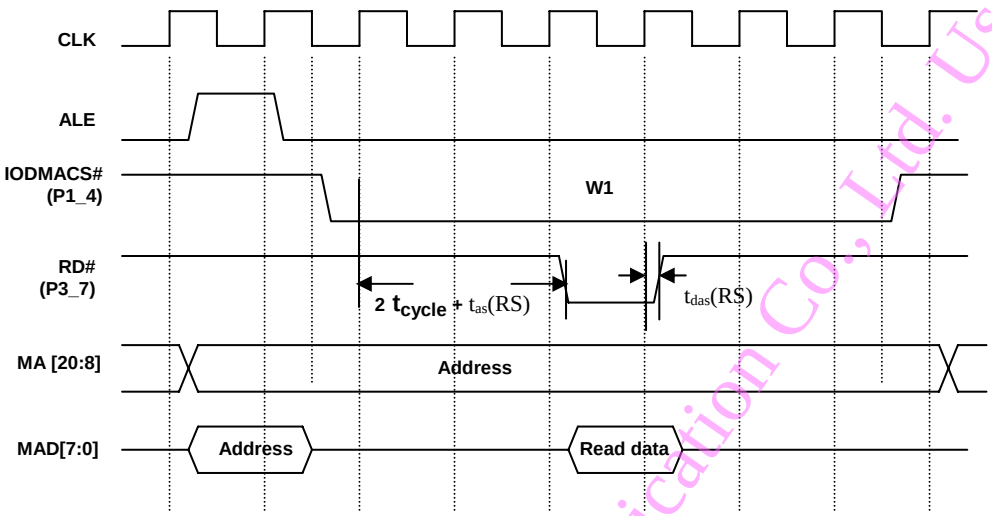
5.3.12 8-bit IO DMA Read Using Wait State Control (3 wait states)



Item	Symbol	Min.	Max.	Unit
IODMA Chip Select assert time	$t_{as(IODCS)}$	-	12	ns
IODMA Chip Select de-assert time	$t_{das(IODCS)}$	-	12	ns
IODMA Read Strobe assert time	$t_{as(RS)}$	-	12	ns
IODMA Read Strobe de-assert time	$t_{das(RS)}$	-	12	ns
Read Data Input data setup time	$t_{s(RD)}$	5	-	ns
Read Data Input data hold time	$t_{h(RD)}$	2	-	ns

5.3.13 8-bit IO DMA Read Using Wait State Control (3 wait states)

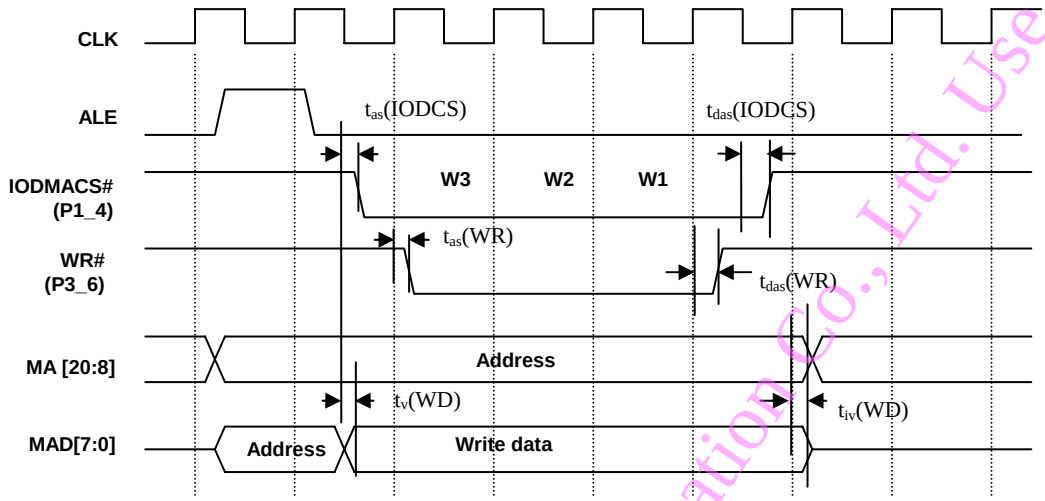
In the timing diagram, the access wait state is 1. And the IODMACS# falling to RD# falling, RD# rising to IODMACS# rising delay are 2 clocks.



Item	Symbol	Min.	Max.	Unit
IODMA Read Strobe assert time	$t_{as}(RS)$	-	12	ns
IODMA Read Strobe de-assert time	$t_{das}(RS)$	-	12	ns

Note: t_{cycle} is the cycle time of the system clock

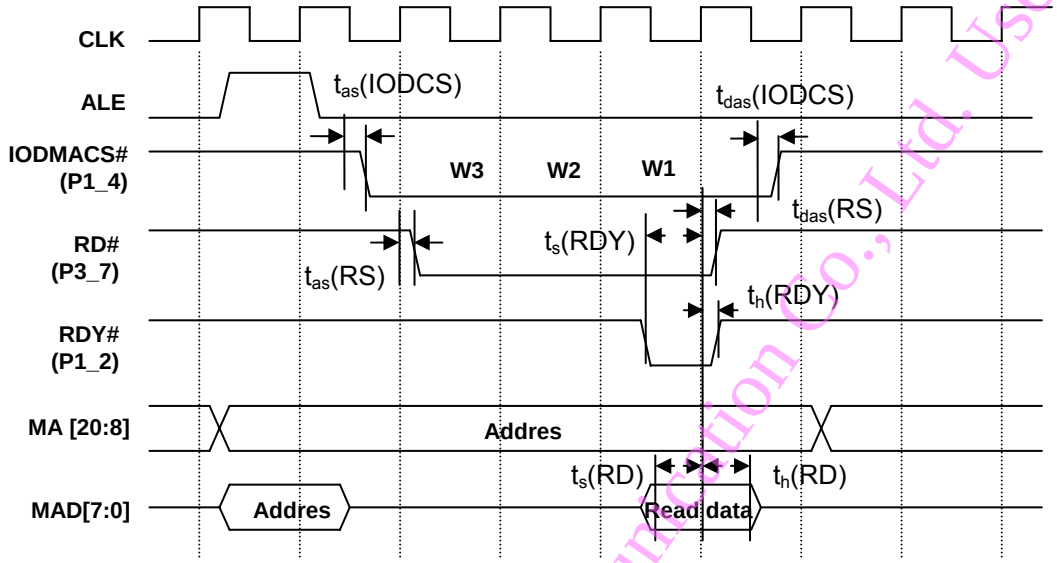
5.3.14 8-bit IO DMA Write Using Wait State Control (3 wait states)



Item	Symbol	Min.	Max.	Unit
IODMA Chip Select assert time	$t_{as(IODCS)}$	-	12	ns
IODMA Chip Select de-assert time	$t_{das(IODCS)}$	-	12	ns
IODMA Write Strobe assert time	$t_{as(WR)}$	-	12	ns
IODMA Write Strobe de-assert time	$t_{das(WR)}$	-	12	ns
Write Data valid time	$t_{v(WD)}$	-	12	ns
Write Data invalid time	$t_{iv(WD)}$	-	12	ns

5.3.15 8-bit IO DMA Read Using External RDY Control

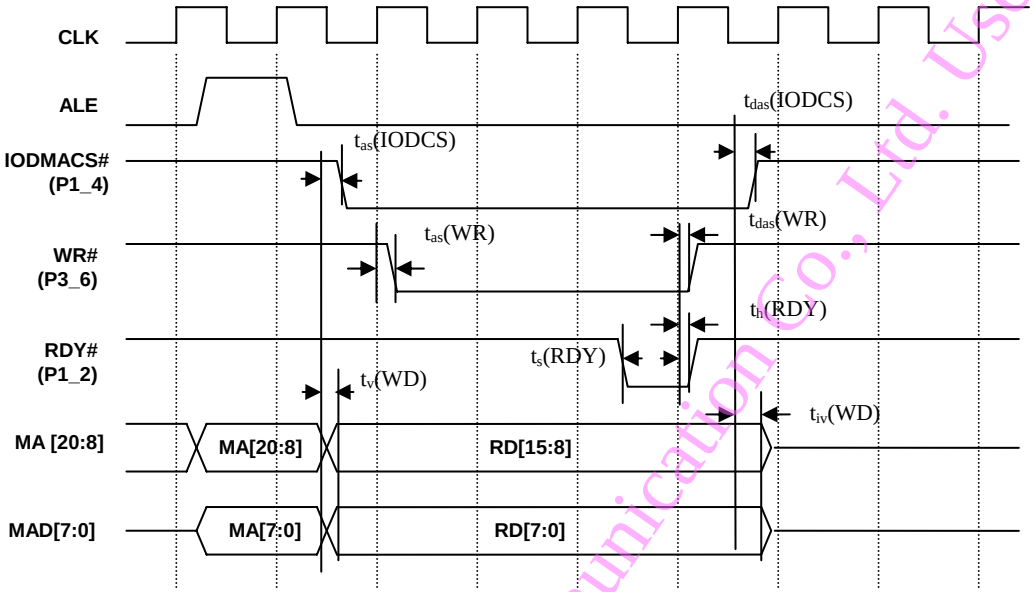
In this timing diagram, the access wait state is 3. The IOCS# falling to RD# falling, RD# rising to IOCS# rising delay are no effect for this access.



Item	Symbol	Min.	Max.	Unit
IODMA Chip Select assert time	t_{as}(IODCS)	-	12	ns
IODMA Chip Select de-assert time	t_{das}(IODCS)	-	12	ns
IODMA Read Strobe assert time	t_{as}(RS)	-	12	ns
IODMA Read Strobe de-assert time	t_{das}(RS)	-	12	ns
RDY input setup time	t_{s}(RDY)	5	-	ns
RDY input hold time	t_{h}(RDY)	2	-	ns
Read Data Input data setup time	t_{S}(RD)	5	-	ns
Read Data Input data hold time	t_{h}(RD)	2	-	ns

5.3.16 16-bit IODMA Write Using External RDY Control

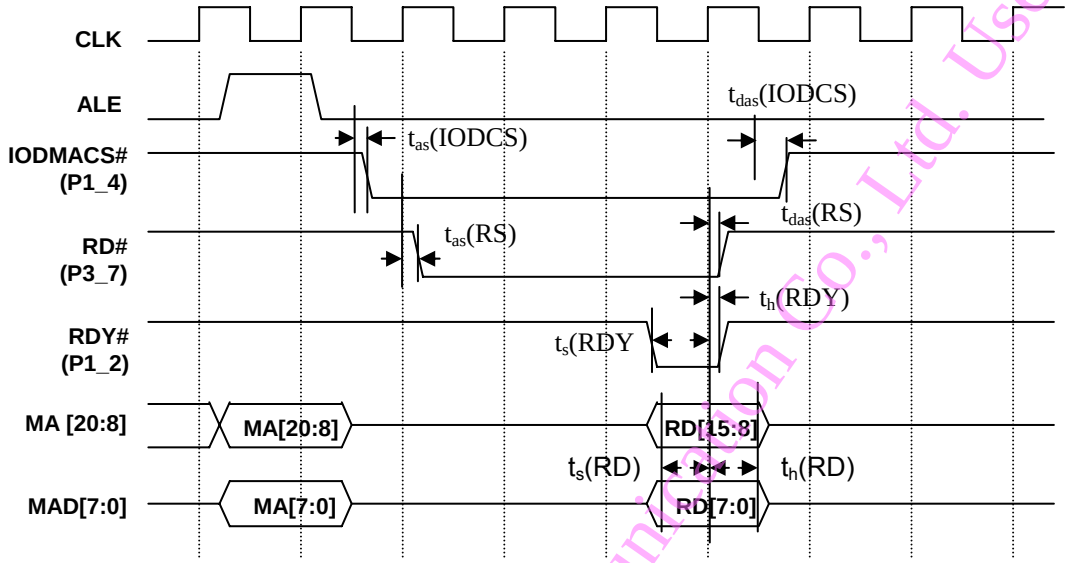
The IOCS# falling to RD# falling, and RD# rising to IOCS# rising delay are not support for 16-bit IODMA. And this timing diagram is for little endian mode.



Item	Symbol	Min.	Max.	Unit
IODMA Chip Select assert time	t_{as}(IODCS)	-	12	ns
IODMA Chip Select de-assert time	t_{das}(IODCS)	-	12	ns
IODMA Write Strobe assert time	t_{as}(WR)	-	12	ns
IODMA Write Strobe de-assert time	t_{das}(WR)	-	12	ns
RDY input setup time	t_{s}(RDY)	5	-	ns
RDY input hold time	t_{h}(RDY)	2	-	ns
Write Data valid time	t_{v}(WD)	-	12	ns
Write Data invalid time	t_{iv}(WD)	-	12	ns

5.3.17 16-bit IODMA Read Using External RDY Control

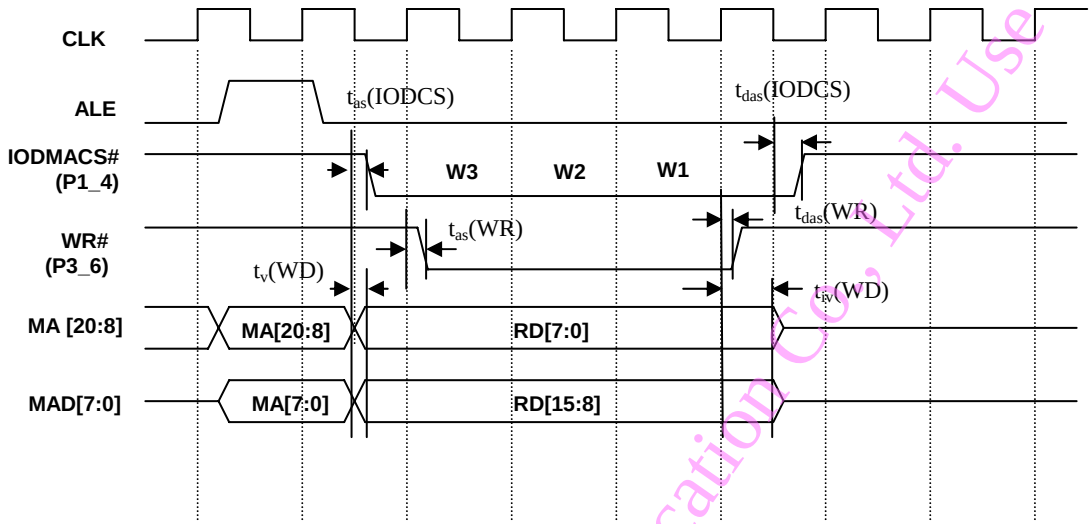
The IOCS# falling to RD# falling, and RD# rising to IOCS# rising delay are not support for 16-bit IODMA. And this timing diagram is for little endian mode.



Item	Symbol	Min.	Max.	Unit
IODMA Chip Select assert time	t_{as}(IODCS)	-	12	ns
IODMA Chip Select de-assert time	t_{das}(IODCS)	-	12	ns
IODMA Read Strobe assert time	t_{as}(RS)	-	12	ns
IODMA Read Strobe de-assert time	t_{das}(RS)	-	12	ns
RDY input setup time	t_{s}(RDY)	5	-	ns
RDY input hold time	t_{h}(RDY)	2	-	ns
Read Data Input data setup time	t_{S}(RD)	5	-	ns
Read Data Input data hold time	t_{h}(RD)	2	-	ns

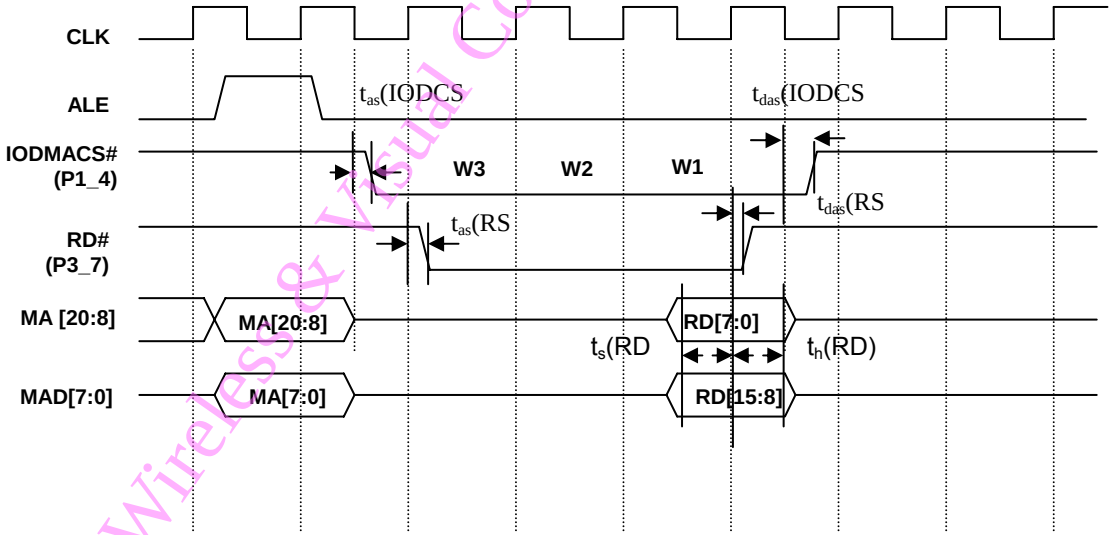
5.3.18 16-bit IODMA Write Using Wait State Control (3 wait states)

The timing diagram is for big endian mode.

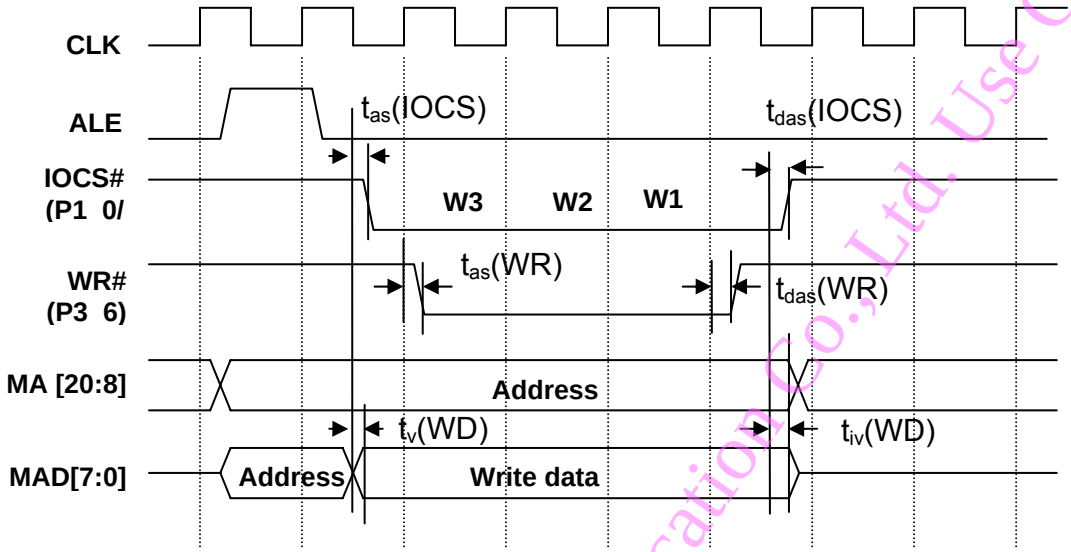


5.3.19 16-bit IODMA Read Using Wait State Control (3 wait states)

The timing diagram is for big endian mode.

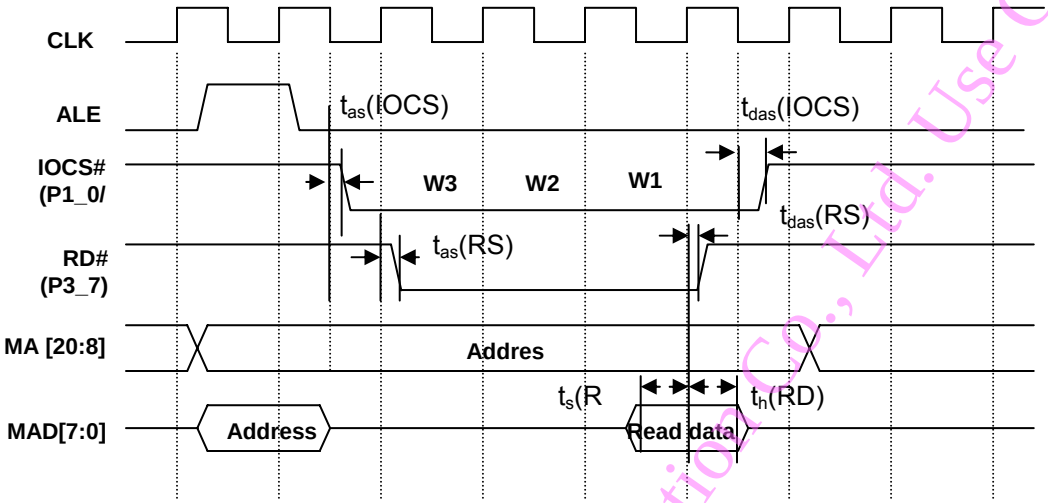


5.3.20 8-bit Extended IO Write Using Wait State Control (3 wait states)



Item	Symbol	Min.	Max.	Unit
IODMA Chip Select assert time	$t_{as(IOCS)}$	-	12	ns
IODMA Chip Select de-assert time	$t_{das(IOCS)}$	-	12	ns
IODMA Write Strobe assert time	$t_{as(WR)}$	-	12	ns
IODMA Write Strobe de-assert time	$t_{das(WR)}$	-	12	ns
Write Data valid delay time	$t_{v(WD)}$	-	12	ns
Write Data invalid delay time	$t_{iv(WD)}$	-	12	ns

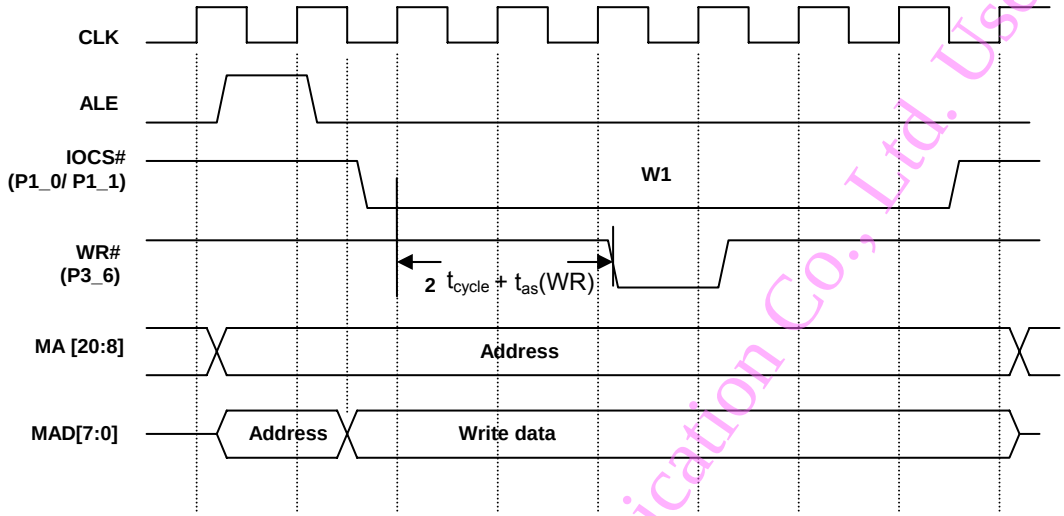
5.3.21 8-bit Extended IO Read Using Wait State Control



Item	Symbol	Min.	Max.	Unit
IODMA Chip Select assert time	t_{as}(IOCS)	-	12	ns
IODMA Chip Select de-assert time	t_{das}(IOCS)	-	12	ns
IODMA Read Strobe assert time	t_{as}(RS)	-	12	ns
IODMA Read Strobe de-assert time	t_{das}(RS)	-	12	ns
RDY input setup time	t_{s}(RDY)	5	-	ns
RDY input hold time	t_{h}(RDY)	2	-	ns
Read Data Input data setup time	t_{s}(RD)	5	-	ns
Read Data Input data hold time	t_{h}(RD)	2	-	ns

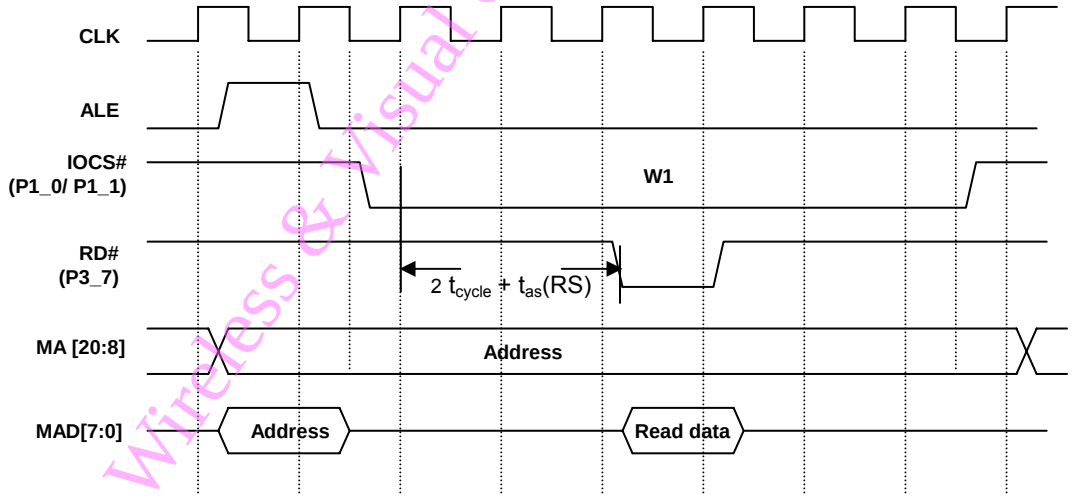
5.3.22 8-bit Extended IO Write Using Wait State Control with Delay Control

In this timing diagram, the wait state is 1, and IOCS# falling to WR# falling and WR# rising to IOCS# rising delay is 2 system clocks.

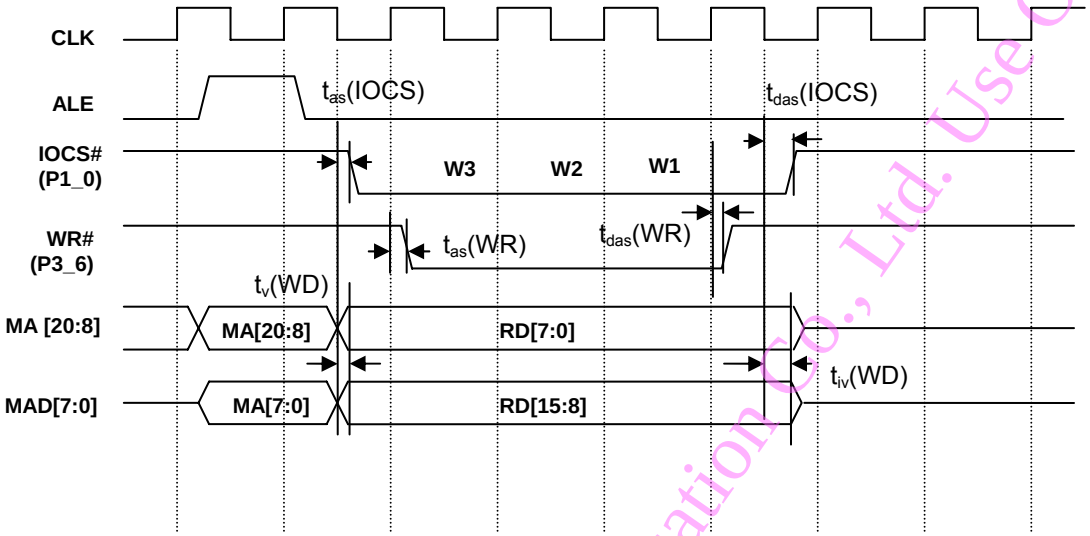


5.3.23 8-bit Extended IO Read Using Wait State Control with Delay Control

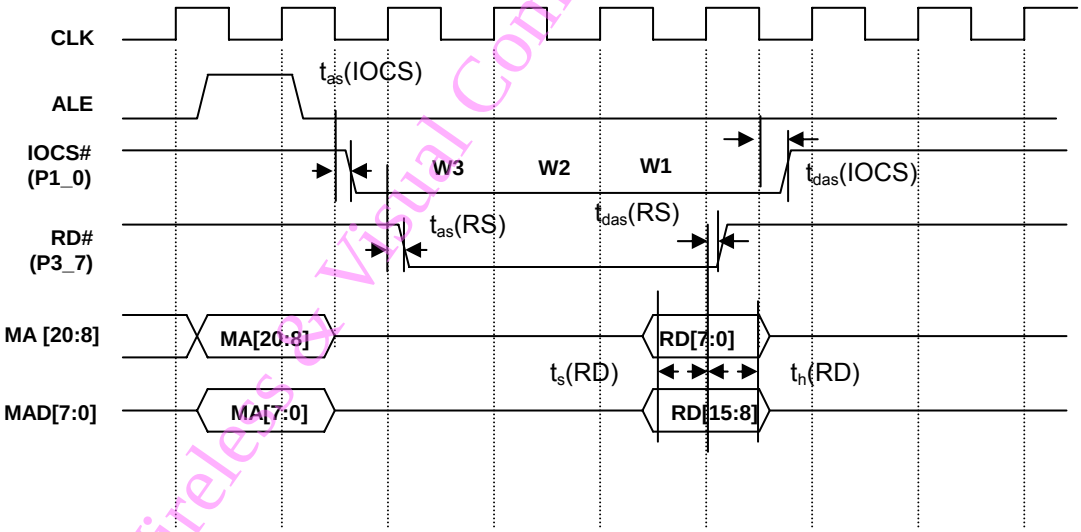
In this timing diagram, the wait state is 1, and IOCS# falling to WR# falling and WR# rising to IOCS# rising delay is 2 system clocks.



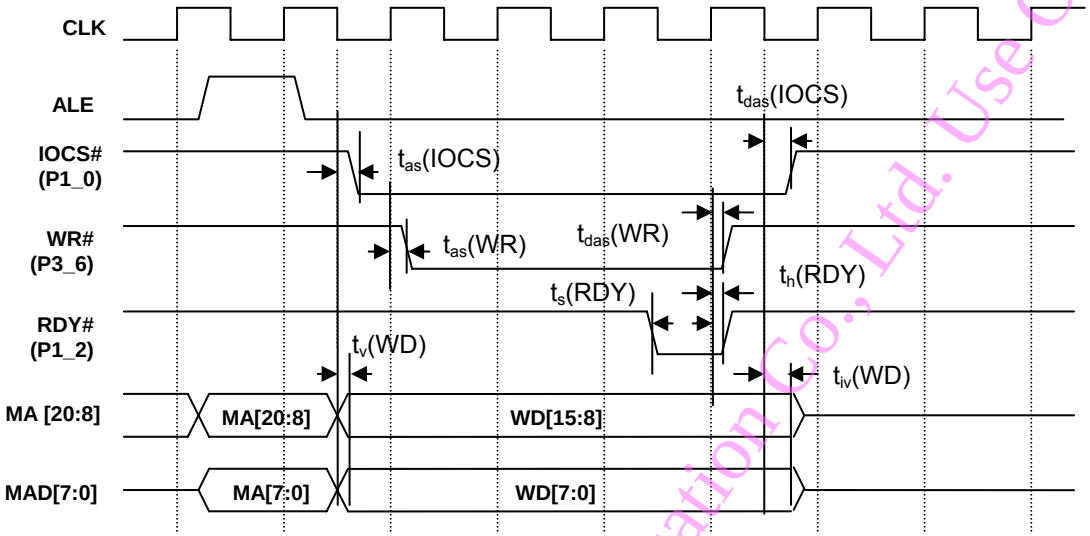
5.3.24 16-bit Extended IO0 Write Using Wait State Control



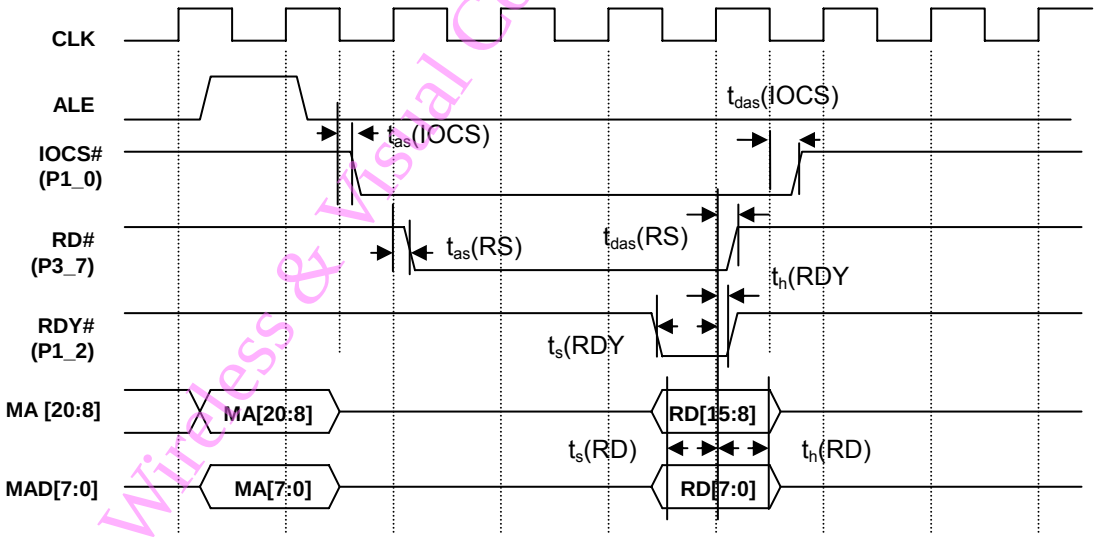
5.3.25 16-bit Extended IO0 Read Using Wait State Control



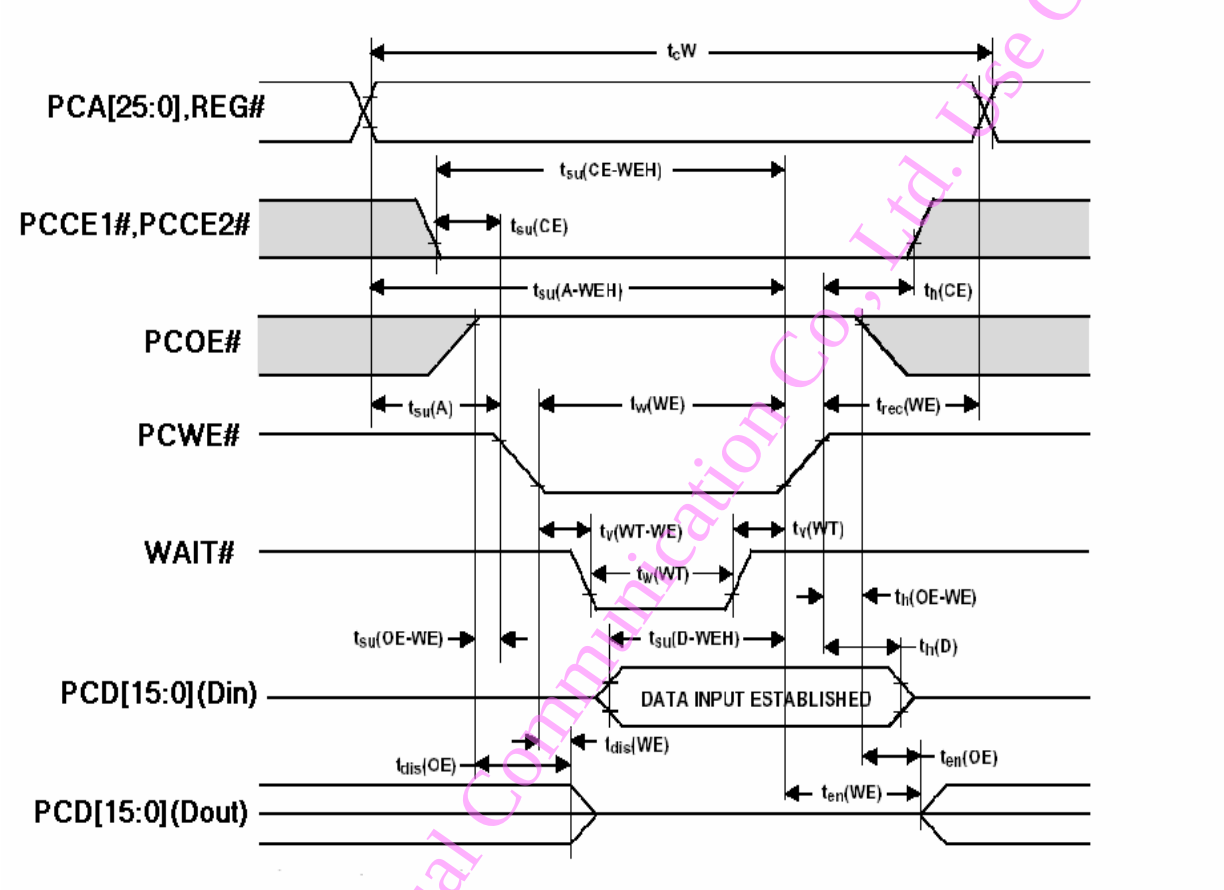
5.3.26 16-bit Extended IO0 Write Using External RDY Control



5.3.27 16-bit Extended IO0 Read Using External RDY Control

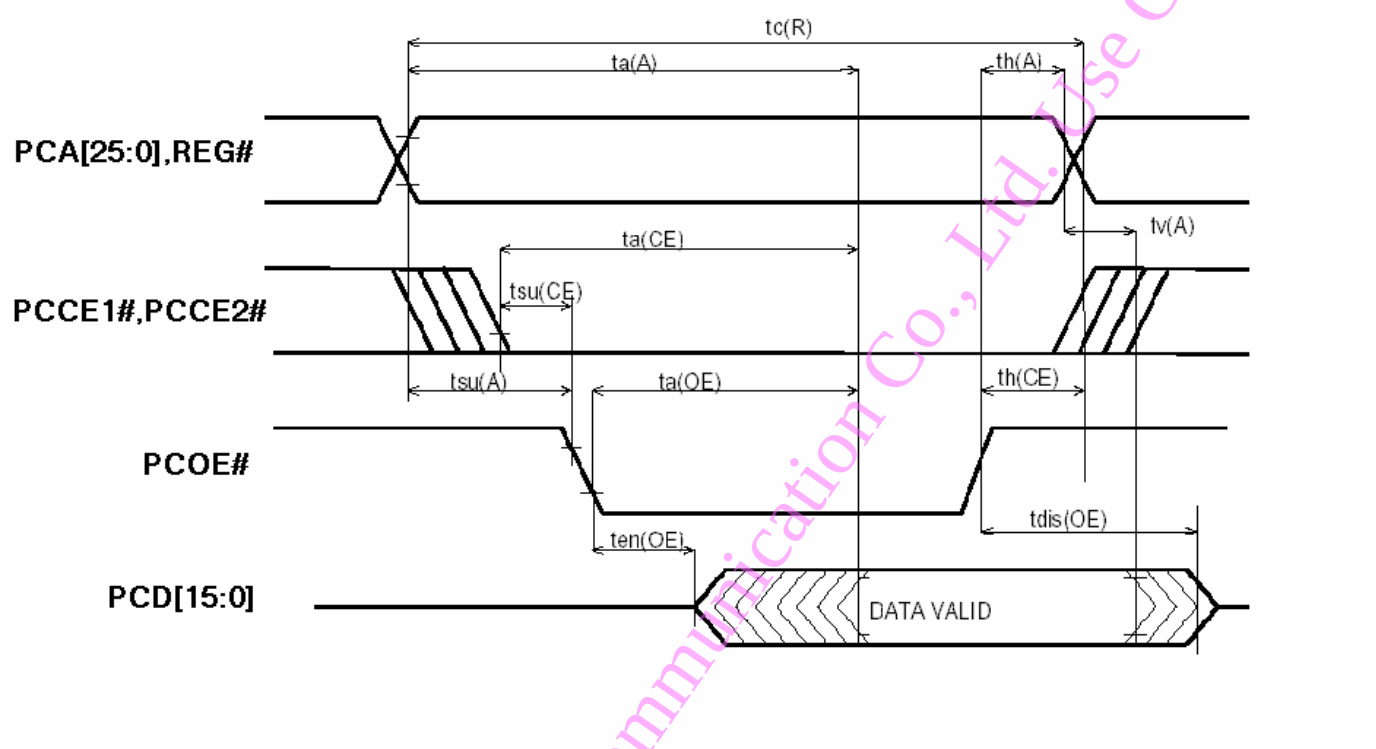


5.3.28 PCMCIA Memory Write Timing



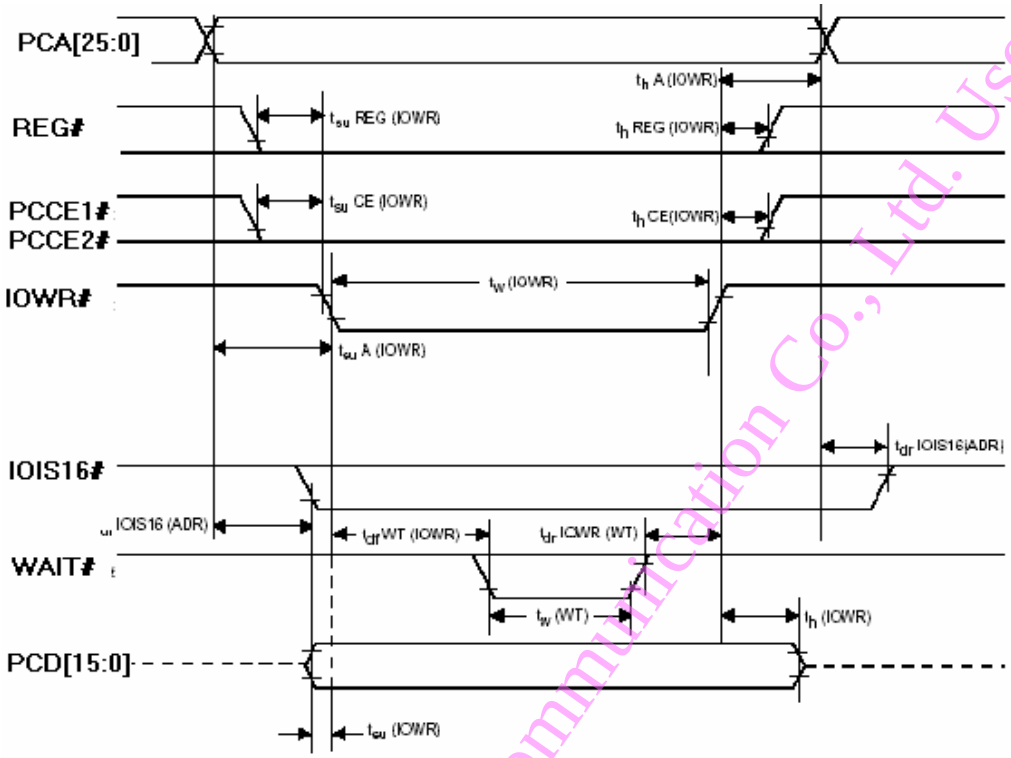
Item	Symbol	Min.	Max.	Unit
Write Cycle Time	t_cW	600		ns
Write Pulse Width	t_w(WE)	300		ns
Address Setup Time	t_{su}(A)	50		ns
Address Setup Time for WE#	t_{su}(A-WEH)	350		ns
Card Enable Setup Time for WE#	t_{su}(CE-WEH)	300		ns
Data Setup Time for WE#	t(D-WEH)	150		ns
Data Hold Time	t_h(D)	70		ns
Write Recover Time	t_{rec}(WE)	70		ns
Output Disable Time from WE#	t_{dis}(WE)		150	ns
Output Disable Time from OE#	t_{dis}(OE)		150	ns
Output Enable Time from WE#	t_{en}(WE)	5		ns
Output Enable Time from OE#	t_{en}(OE)	5		ns
Output Enable Setup from WE#	t_{su}(OE-WE)	35		ns
Output Enable Hold from WE#	t_h(OW-WE)	35		ns
Card Enable Setup Time	t_{su}(CE)	0		ns
Card Enable Hold Time	t_h(CE)	35		ns
WAIT# Valid from WE#	t_v(WT-WE)		100	ns
WAIT# Pulse Width	t_w(WT)		12us	ns
WE# High from WAIT# Released	t_v(WT)	0		ns

5.3.29 PCMCIA Memory Read Timing



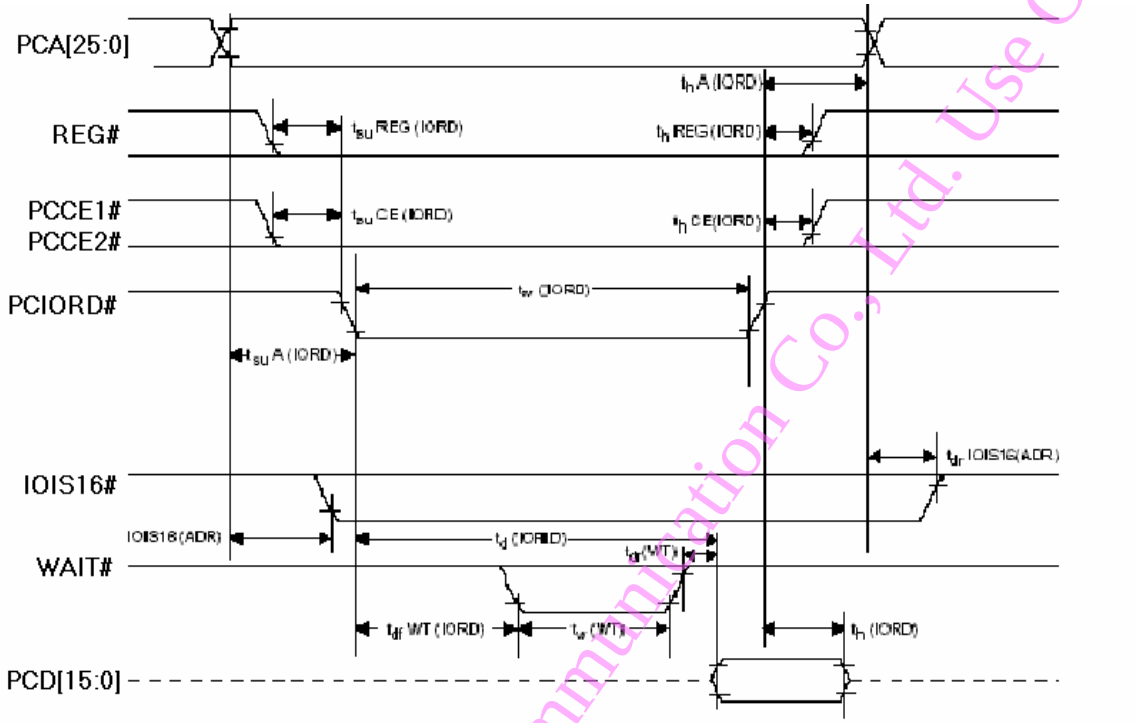
Item	Symbol	Min	Max	Unit
Read Cycle Time	t_c(R)	600		ns
Address Access Time	t_a(A)		600	ns
Card Enable Access Time	t_a(CE)		600	ns
Output Enable Access Time	t_a(OE)		300	ns
Output Disable Time from OE#	t_{dis}(OE)		150	ns
Output Enable Time from OE#	t_{en}(OE)	5		ns
Data Valid from Add Change	t_v(A)	0		ns
Address Setup Time	t_{su}(A)	100		ns
Address Hold Time	t_h(A)	35		ns
Card Enable Setup Time	t_{su}(CE)	0		ns
Card Enable Hold Time	t_h(CE)	35		ns
Output Enable Time from OE#	t_{en}(OE)	5		ns

5.3.30 PCMCIA I/O Write Timing



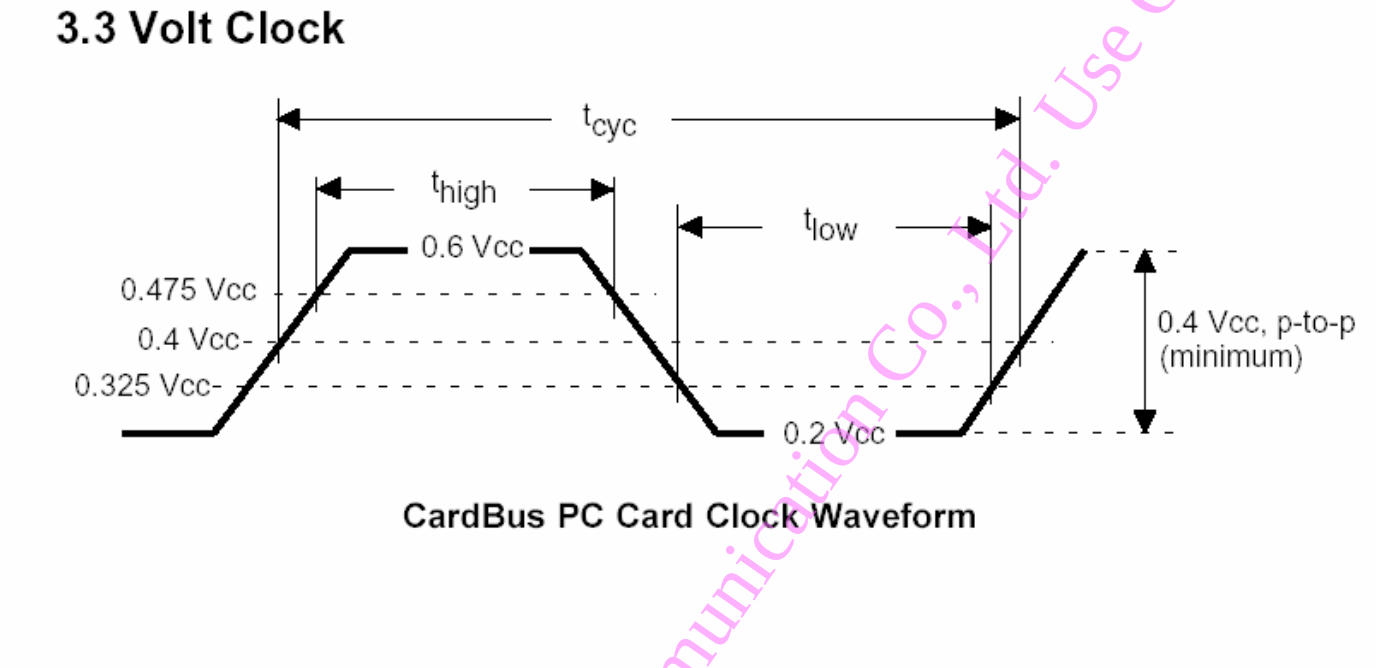
Item	Symbol	Min	Max	Unit
Data Delay after IOWR#	t_d(IOWR)	60		ns
Data Hold following IOWR#	t_h(IOWR)	30		ns
IOWR# Width Time	t_w(IOWR)	165		ns
Address Setup before IOWR#	t_{su} A(IOWR)	70		ns
Address Hold following IOWR#	t_h A(IOWR)	20		ns
CE# Setup before IOWR#	t_{su} CE(IOWR)	5		ns
CE# Hold following IOWR#	t_h CE(IOWR)	20		ns
REG# Setup before IOWR#	t_{su}REG(IOWR)	5		ns
REG# Hold following IOWR#	t_h REG(IOWR)	0		ns
IOIS16# Delay Falling from Address	t_{df}IOIS16(ADR)		35	ns
IOIS16# Delay Rising from Address	t_{dr}IOIS16(ADR)		35	ns
WAIT# Delay Fallling from IOWR#	t_{df}WT(IOWR)		35	ns
WAIT# Width Time	t_w(WT)		12,000	ns
IOWR# high from WAIT# high	t_{dr}IOWR(WT)	0		ns

5.3.31 PCMCIA I/O Read Timing



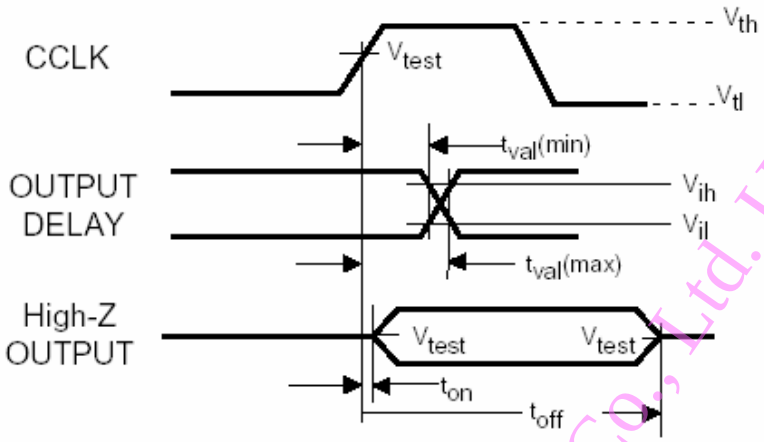
Item	Symbol	Min	Max	Unit
Data Delay after IORD#	t_d(IORD)		100	ns
Data Hold following IORD#	t_h(IORD)	0		ns
IORD# Width Time	t_w(IORD)	165		ns
Address Setup before IORD#	t_{su} A(IORD)	70		ns
Address Hold following IORD#	t_h A(IORD)	20		ns
CE# Setup before IORD#	t_{su} CE(IORD)	5		ns
CE# Hold following IORD#	t_h CE(IORD)	20		ns
REG# Setup before IORD#	t_{su}REG(IORD)	5		ns
REG# Hold following IORD#	t_h REG(IORD)	0		ns
IOIS16# Delay Falling from Address	t_{df}IOIS16(ADR)		35	ns
IOIS16# Delay Rising from Address	t_{dr}IOIS16(ADR)		35	ns
WAIT# Delay Fallling from IORD#	t_{df}WT(IORD)		35	ns
Data Delay from WAIT# Rising	t_{dr}(WT)		35	ns
WAIT# Width Time	t_w(WT)		12,000	ns

5.3.32 CardBus PC Card Clock Specifications



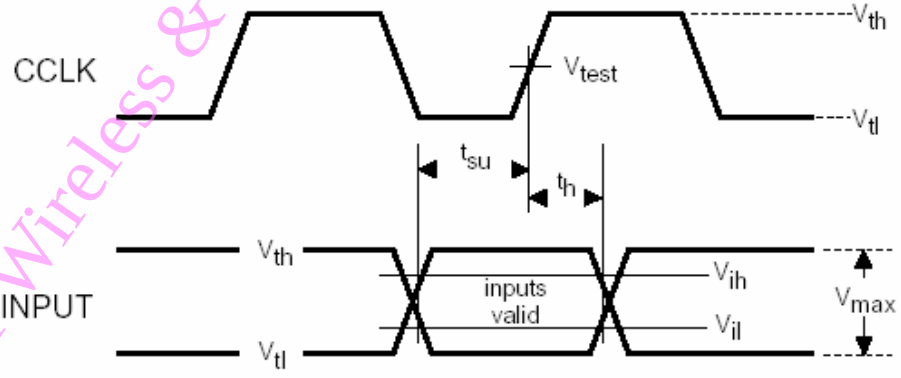
CardBus PC Card Clock Waveform

Item	Symbol	Min.	Max.	Unit
CCLK Cycle Time	t_{cyc}	30	∞	**ns**
CCLK High Time	t_{high}	12	-	**ns**
CCLK Low Time	t_{low}	12	-	**ns**
CCLK Slew Rate (0.2Vcc-0.6Vcc)	t_{slew}	1	4	**ns**



Output Timing Measurement Conditions

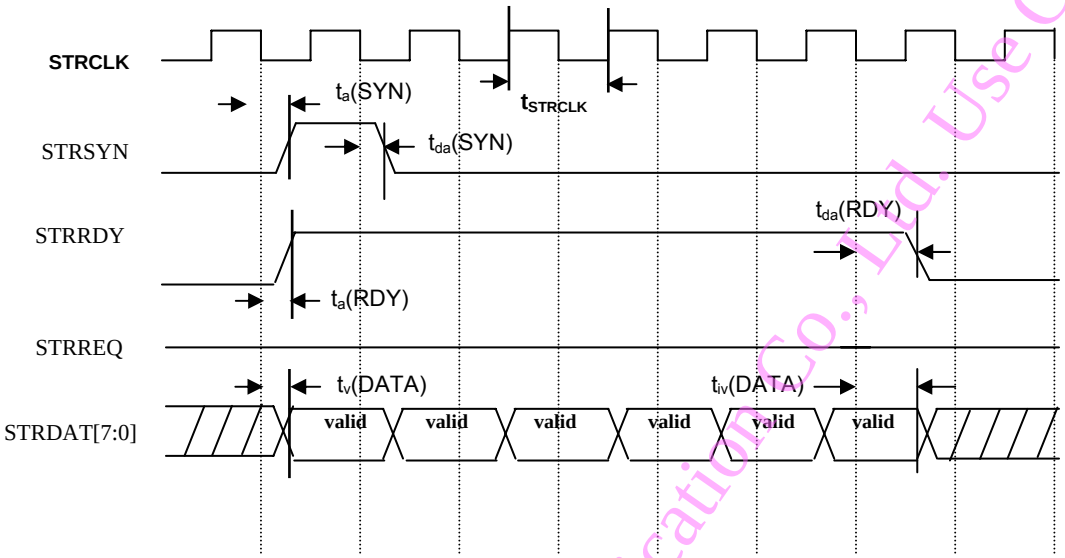
Item	Symbol	Min	Max	Unit
CCLK to Signal Valid Delay	t_{val}	2	18	**ns**
Float to Active Delay	t_{on}	2		**ns**
Active to Float Delay	t_{off}		28	**ns**
Input Setup Time to CCLK	t_{su}	7		**ns**
Input Hold Time to CCLK	t_h	0		**ns**
Reset Active Time After Power Stable	t_{rst}	1ms		**ns**
Reset Active Time After CCLK Stable	$t_{rst-clk}$	100 clocks		**ns**
Reset Active to Output float delay	$t_{rst-off}$		40	**ns**
CSTSCHG remote wakeup pulse width	t_{pulse}	1ms	40	**ns**



Input Timing Measurement Conditions

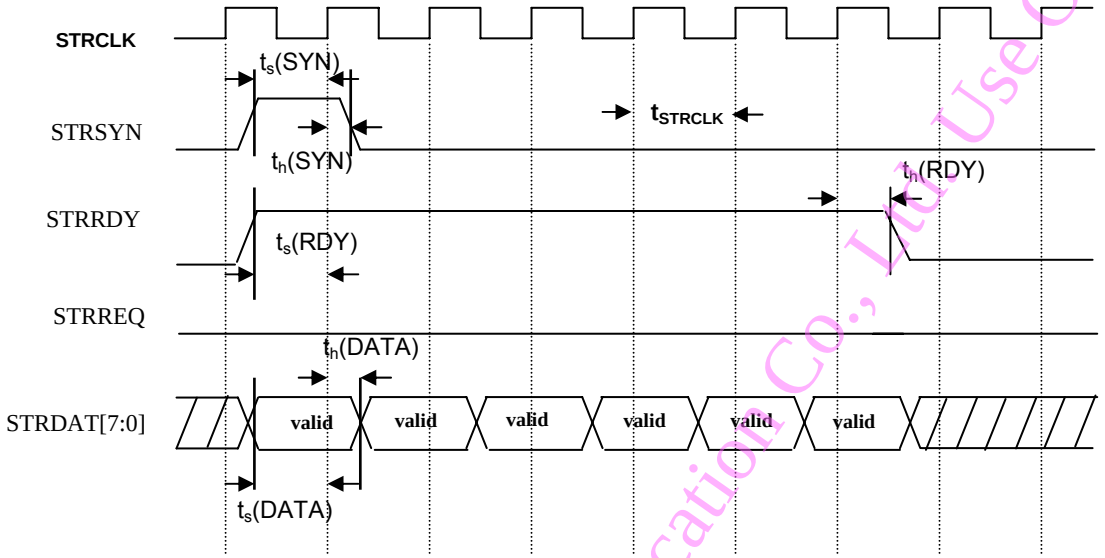
Symbol	3.3V signaling	Unit
V_{th}	0.6Vcc	V
V_{tl}	0.2Vcc	V
V_{ih}	0.475Vcc	V
V_{il}	0.325Vcc	V
V_{test}	0.4Vcc	V
V_{max}	0.4Vcc	V

5.3.33 Video Streaming Interface TS Streaming Output Timing



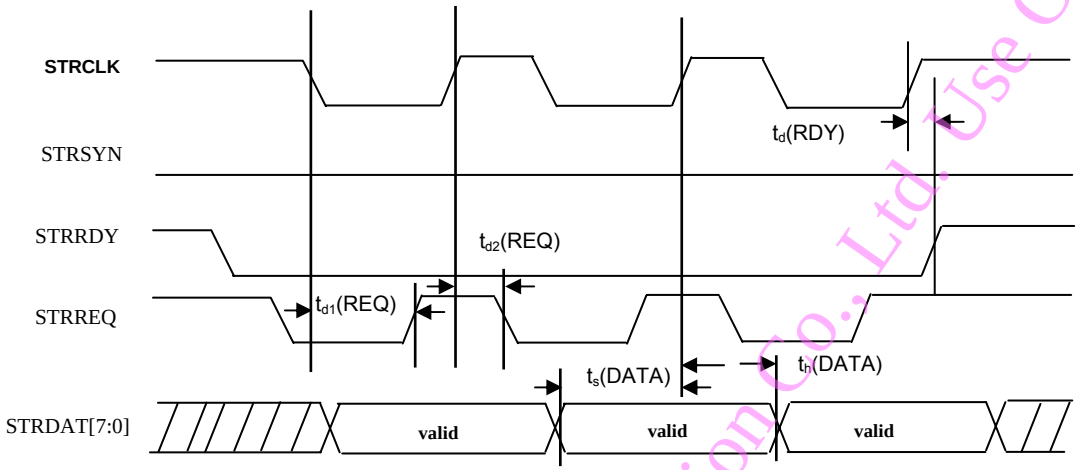
Item	Symbol	Min.	Max.	Unit
Stream output clock cycle time for STRCLK	t_{STRCK}	t_{VIDEO_CLK} x3	t_{VIDEO_CLK} x255	ns
Stream Sync. assert delay time after STRCLK falling edge	$t_{a(SYN)}$	-	2	ns
Stream Sync. de-asset delay time after STRCLK falling edge	$t_{da(SYN)}$	-	2	ns
Stream RDY assert delay time after STRCLK falling edge	$t_{a(RDY)}$	-	2	ns
Stream RDY de-assert delay time after STRCLK falling edge	$t_{da(RDY)}$	-	2	ns
Stream Data valid delay time after STRCLK falling edge	$t_{V(DATA)}$	-	2	ns
Stream Data invalid delay time after STRCLK falling edge	$t_{iv(DATA)}$	-	2	ns

5.3.34 Video Streaming Interface TS Streaming Input Timing



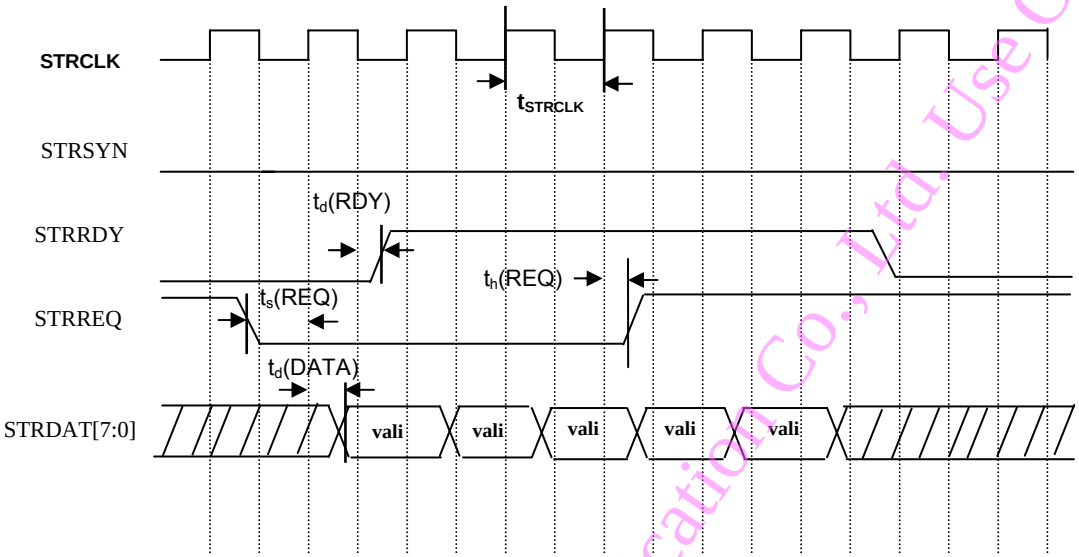
Item	Symbol	Min.	Max.	Unit
Stream input clock cycle time	t_{STRCK}	t_{SYSCLK} x4	-	ns
Stream Sync. input setup time	$t_{s(SYN)}$	t_{SYSCLK} x2	-	ns
Stream Sync. input hold time	$t_{h(SYN)}$	t_{SYSCLK}	-	ns
Stream RDY input setup time	$t_{s(RDY)}$	t_{SYSCLK} x3	-	ns
Stream RDY input hold time	$t_{h(RDY)}$	t_{SYSCLK}	-	ns
Stream Data input setup time	$t_{s(DATA)}$	2	-	ns
Stream Data input hold time	$t_{h(DATA)}$	0.5	-	ns

5.3.35 Video Streaming Interface PS Byte Transfer Mode Input Timing



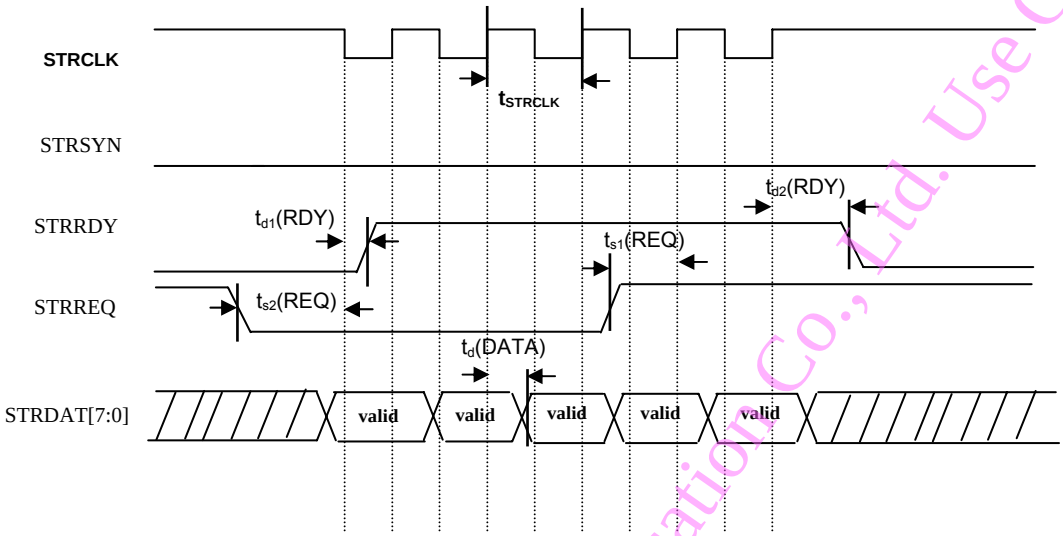
Item	Symbol	Min.	Max.	Unit
Stream request falling edge output delay time after STRRDY falling edge	$t_{d(RDY)}$	-	t_{SYSCLK} x3	ns
Stream request rising edge output delay time after STRCLK falling edge	$t_{d1(REQ)}$	-	t_{SYSCLK} x6	ns
Stream request falling edge output delay time after STRCLK rising edge	$t_{d2(REQ)}$	-	t_{SYSCLK} x22	ns
Stream data input setup time	$t_{s(DATA)}$	1.5	-	ns
Stream data input hold time	$t_{h(DATA)}$	0.5	-	ns

5.3.36 Video Streaming Interface PS Mode Output Timing



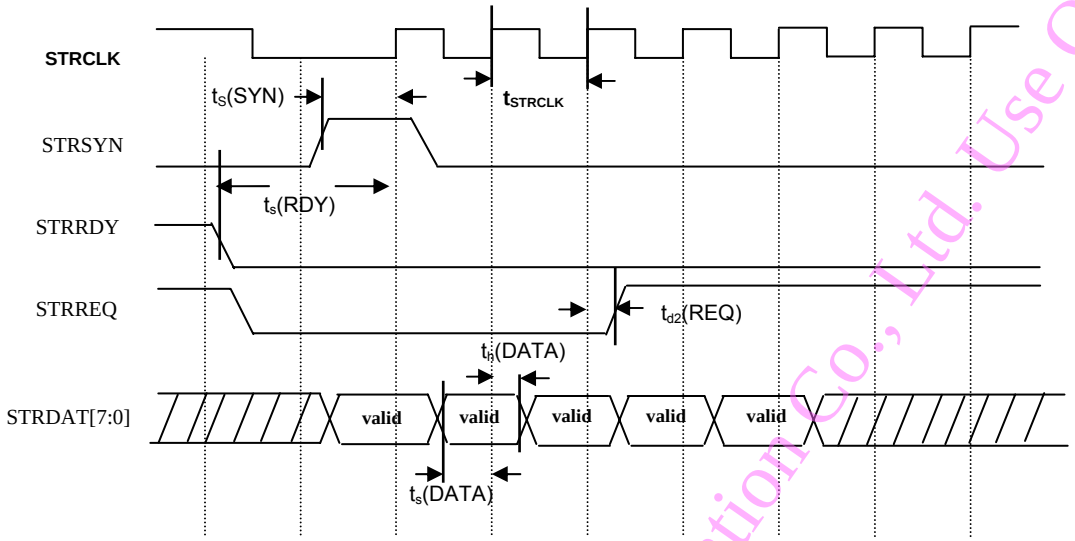
Item	Symbol	Min.	Max.	Unit
Stream output clock cycle time for NEC ISCLK	t_{STRCLK}	t_{VIDEO_CLK} x3	t_{VIDEO_CLK} x255	ns
Stream RDY output delay time after STRCLK falling edge	$t_{d(RDY)}$	-	10	ns
Stream request input setup time require before STRCLK rising edge	$t_{s(REQ)}$	t_{VIDEO_CLK} x3	-	ns
Stream request input hold time require after STRCLK rising edge	$t_{h(REQ)}$	t_{VIDEO_CLK}	-	ns
Stream data output delay time after STRCLK rising edge	$t_{d(DATA)}$	-	t_{SYSCLK} x4	ns

5.3.37 Video Streaming Interface PS Strobe Mode Output Timing



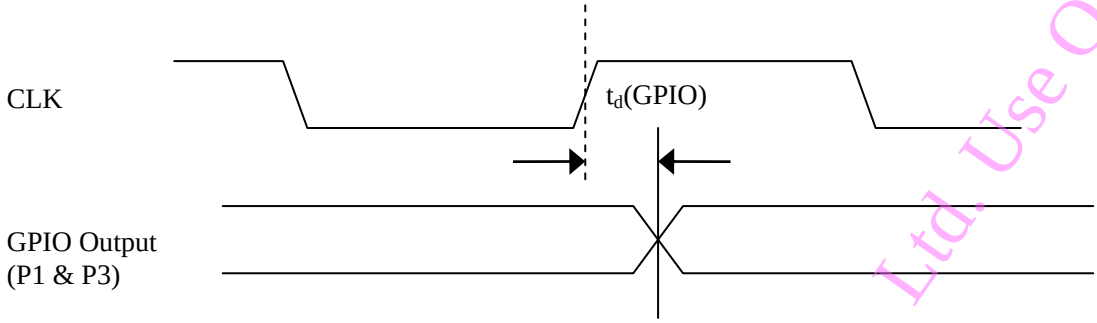
Item	Symbol	Min.	Max.	Unit
Stream output clock cycle time for NEC ISCLK	t_{STRCLK}	t_{VIDEO_CLK} x3	t_{VIDEO_CLK} x255	ns
Stream RDY output delay time after STRCLK falling edge	t_{d1}(RDY)	-	10	ns
Stream RDY output delay time after STRCLK rising edge	t_{d2}(RDY)	t_{d1}(RDY) + t_{STRCLK} / 2		ns
Stream request input setup time require before STRCLK rising edge	t_{s1}(REQ)	t_{VIDEO_CLK} x3	-	ns
Stream request input setup time require before STRCLK falling edge	t_{s2}(REQ)	t_{s1}(REQ) + t_{STRCLK} / 2		ns
Stream data output delay time after STRCLK rising edge	t_{d}(DATA)	-	t_{SYSCLK} x4	ns

5.3.38 Video Streaming Interface PS Strobe Mode Input Timing



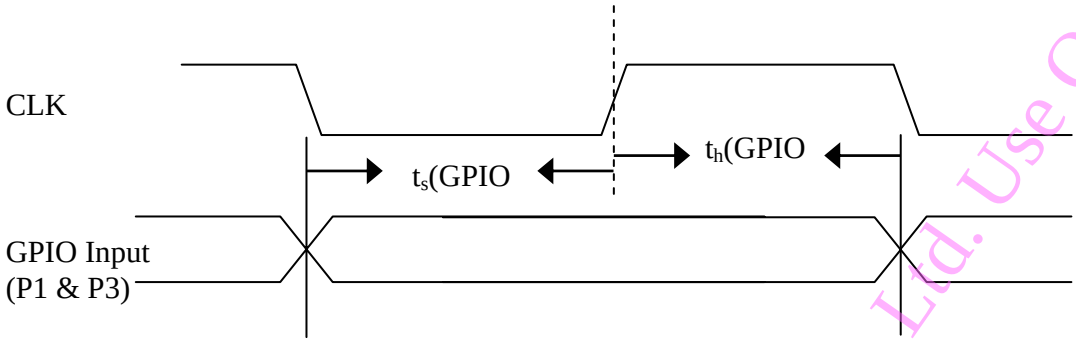
Item	Symbol	Min.	Max.	Unit
Stream input clock cycle time for NEC OSCLK	t_{STRCLK}	t_{VIDEO_CLK} x3	t_{VIDEO_CLK} x255	ns
Sync. input setup time before STRCLK rising edge	t_s(SYN)	t_{SYSCLK} x2	-	ns
STRRDY input setup time before STRCLK rising edge	t_s(RDY)	t_{SYSCLK} x3	-	ns
Stream request rising edge output delay time after STRRDY rising edge	t_d(REQ)	-	t_{SYSCLK} x4	ns
Stream data setup time after STRCLK rising edge	t_s(DATA)	1.5	-	ns
Stream data hold time after STRCLK rising edge	t_h(DATA)	0.5	-	ns

5.3.39 GPIO Output Timing



Item	Symbol	Min.	Max.	Unit
GPIO output high delay time	$t_{d(GPIOH)}$	-	12	ns
GPIO output low delay time	$t_{d(GPIOL)}$	-	12	ns

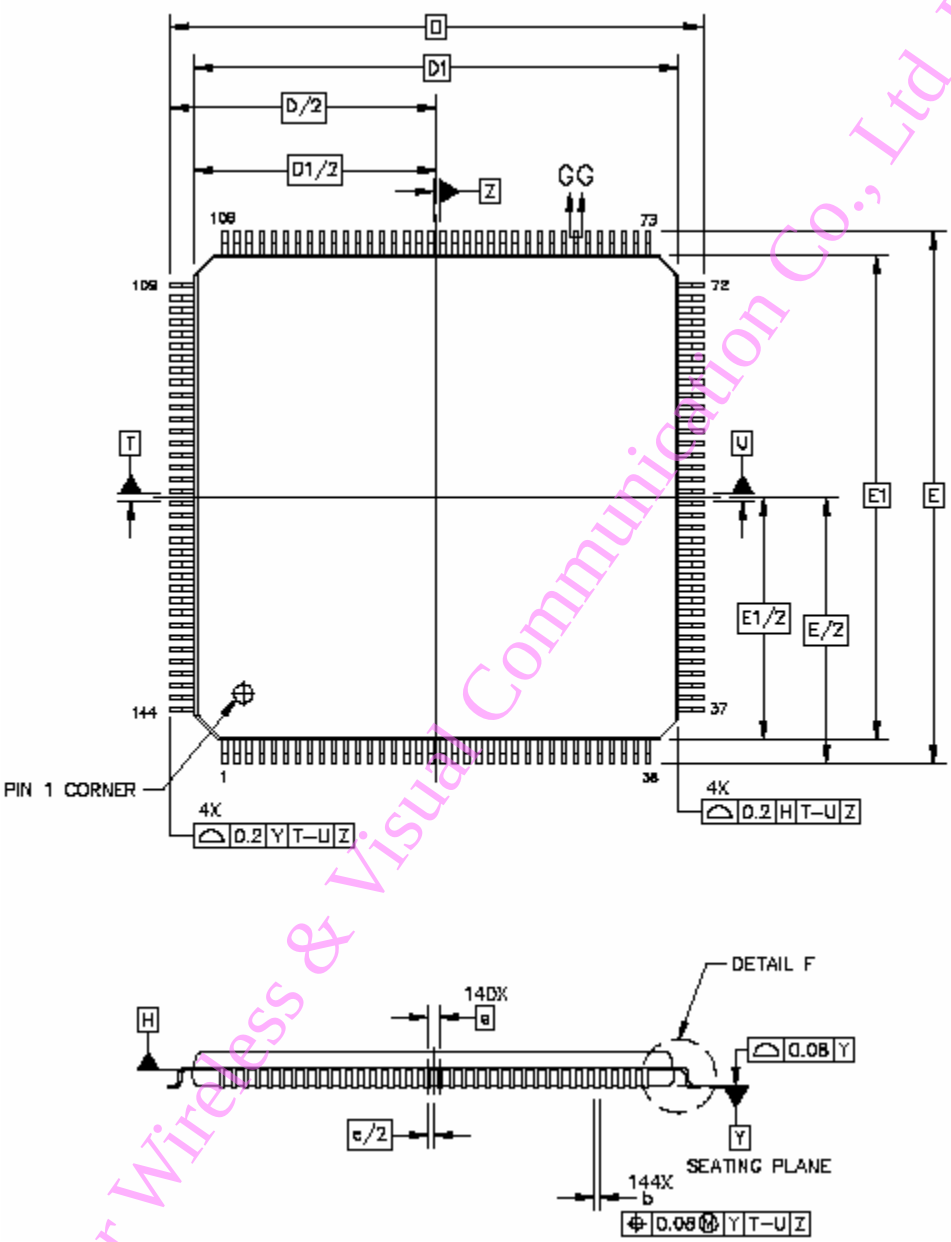
5.3.40 GPIO Input Timing

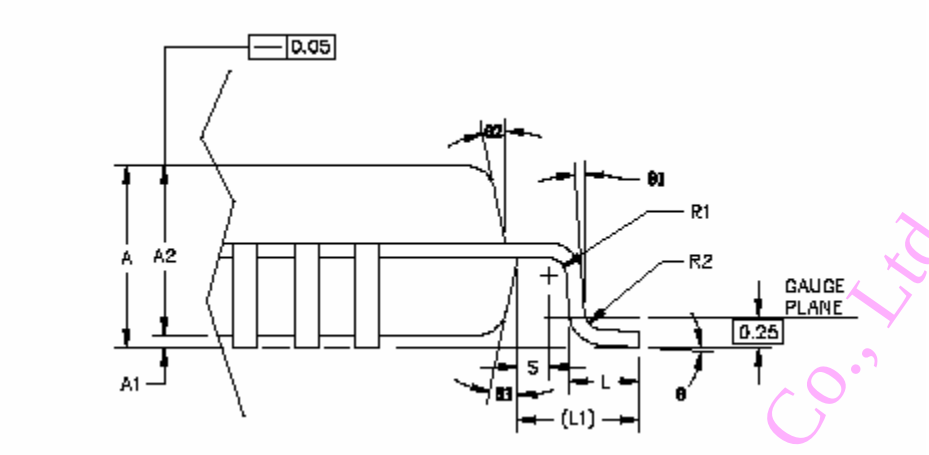


Item	Symbol	Min.	Max.	Unit
GPIO input setup time	$t_{S(GPIO)}$	5	-	ns
GPIO input hold time	$t_{h(GPIO)}$	2	-	ns

6. Mechanical Dimensions

Plastic QFP 144pin with body size 20 x 20 x 14 mm





DIM	MIN	NOM	MAX	DIM	MIN	NOM	MAX
A	–	–	1.6	R1	0.08	–	–
A1	0.05		0.15	R2	0.08	–	0.2
A2	1.35	1.4	1.45	S	0.2	–	–
b	0.17	0.22	0.27	θ	0°	3.5°	7°
b1	0.17	0.2	0.23	θ1	0°	–	–
c	0.09	–	0.2	θ2	11°	12°	13°
c1	0.09	–	0.16	θ3	11°	12°	13°
D		22 BSC					
D1		20 BSC					
e		0.5 BSC					
E		22 BSC					
E1		20 BSC					
L	0.45	0.6	0.75				
L1		1 REF					