Initialization of IDT75T54100 IP Co-Processor

Application Note AN-268

Overview

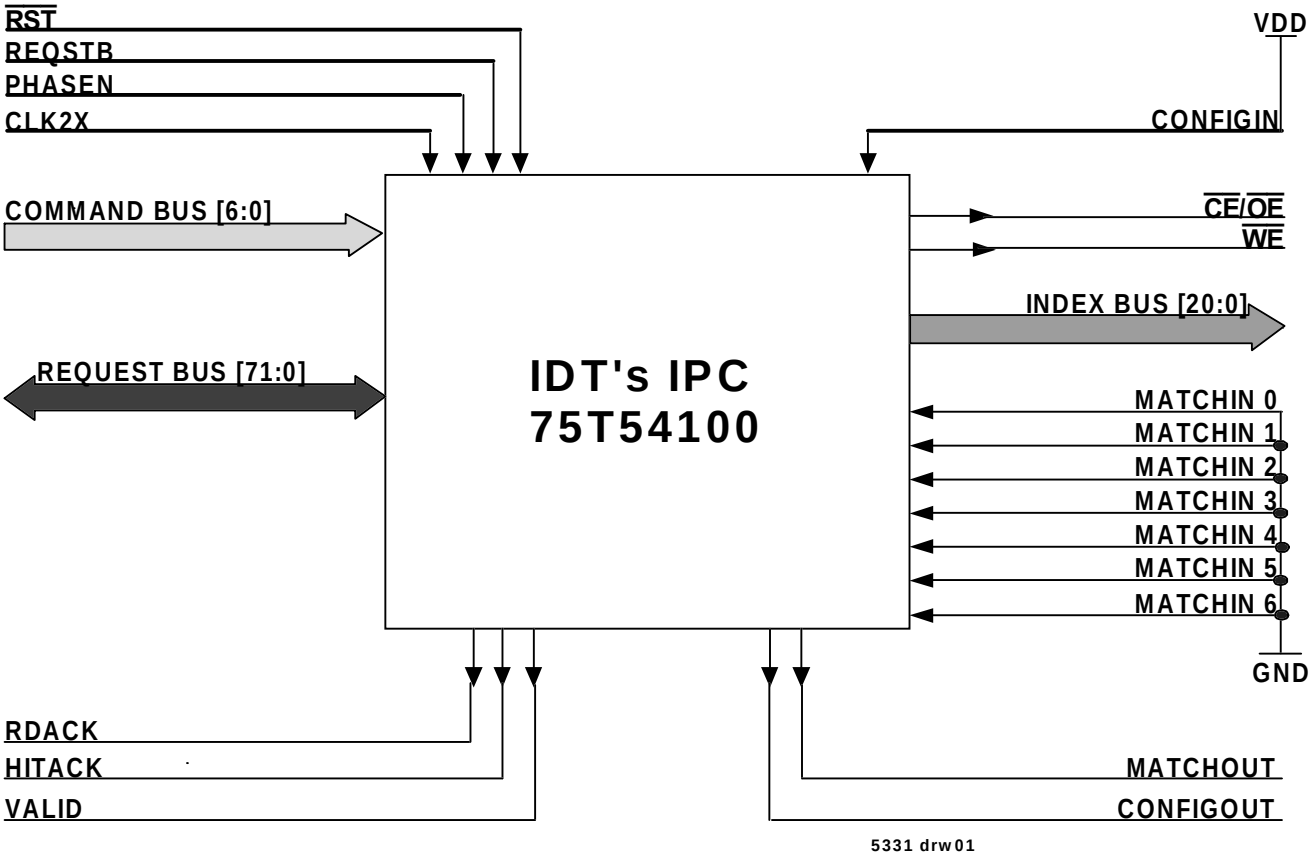
This application note "Initialization of IDT75T54100 IP Co-Processor" device describes how to initialize the IPC from power up. A single IDT75T54100 device can be used or multiple devices can be grouped together for depth expansion. The first part of this application note will discuss how to initialize a single device. It will explain the initialization procedure required in a chronological order and discuss the features that are selectable by the user. The second part of this application note will explain the differences in the initialization sequence required when hooking up multiple devices in a depth expanded IPC system.

Background Information

The IPC (Internet Protocol Co-Processor) family was developed for a wide range of communication and networking applications. The IPC family is intended for use in applications that require high speed data searching such as routers, high layer switching and in the convergence of voice, data, and video.

The IDT75T54100 is part of IPC family of products and is a high performance pipelined, synchronous 64K x 72 IPC. It utilizes content addressable memory technology to perform pattern recognition functions. Each location in the IPC has a Data entry. The IPC has a 72-bit bi-directional bus, which is a 16-bit multiplexed address and 72-bit data bus that can support 66 million lookups per second. Refer to the IDT75T54100 Datasheet for a full device description.

Figure 1.0 Single IPC Device



MAY 2001

Introduction:

◆ Part One: Single Device Initialization

Part One of this applications note describes the procedure to initialize a single IPC device. Figure 1.0 shows the hardware connections for the IDT75T54100 device. The procedure is broken down into the sequences as listed below with each sequence described on its own page.

- ✥ Powering Up a Single Device
- ✥ Background Write of Data Array
- ✥ Initialization of Global Mask Registers (GMRs)
- ✥ Initialization of Reply Width Registers (RWRs)
- ✥ Initialization of System Configuration Register (SCR), Read Only Registers
- ✥ Single Device Summary Table

◆ Part Two: Multiple Device Initialization

Part Two of this applications note describes a general overview how to initialize a depth expanded multiple device IPC system. The differences in the procedure compared to a single device is broken down into the sequences as listed below.

- ✥ Powering Up Multiple Devices
- ✥ Background Write of Data Array
- ✥ Initialization of Global Mask Registers (GMRs)
- ✥ Initialization of Reply Width Registers (RWRs)
- ✥ Initialization of System Configuration Register (SCR)

An example of a multiple device system is described in the following section. Figure 2.0 shows the hardware connections of four IDT75T54100 devices in a depth expanded configuration.

- ✥ Figure 2.0 Example: Multiple IPC Devices
- ✥ Example: Multiple IPC Devices

◆ Powering Up a Single Device

Sequence:

The Reset signal ($\overline{RST}$) must be active (Low) on power up and must remain low while all the power signals (V_{DD}, V_{DDQ}, V_{MATCH}) and the clock signals (CLK2X, PHASEN) become stable. The V_{DD} supply need to begin ramping up first, followed by the V_{DDQ} supply, and the by the V_{MATCH} supply. For power down reverse this order. The IPC will respond to the reset by asynchronously tri-stating the I/O's and output pins which prevents bus contention. The internal logic and System Configuration Register comes out of reset synchronously after the clock signals stabilize and V_{MATCH}, V_{DD} and V_{DDQ} supplies ramp to operating levels. The Enable (En) bit in the System Configuration Register is reset to "0".

Device Identification:

After the power supplies and clock signals have stabilized, the IPC requires that the $\overline{RST}$ and REQSTB signals be low and the clock signals be active for a minimum of thirty-two CLK2X clock cycles to insure proper initialization. Next deactivate the $\overline{RST}$ signal to commence IPC operations. Set the CONFIGIN signal high for a minimum of sixteen CLK2X clock cycles to set the Device ID in the Depth Expansion Register. Once the sixteen CLK2X cycles have passed the IPC is ready to begin initialization procedures.

ASIC/FPGA Handshaking of signals:

The user must initially set the LC bit in the System Configuration Register to enable the handshaking of signals back to the ASIC/FPGA. This enables the RDACK, HITACK and VALID signals to be driven.

Set the REQSTB signal High for two CLK2X cycles to signal the start of a valid IPC operation and do the following: On the Command Bus select the Write instruction, set the CMD bits [3:0] to "0100" and zeros on the rest of the Command Bus. On the Request Bus, select the System Configuration Register, set the Request Bus bits [8:1] to "0000 0011" and zeros to the rest of the Command bus bits.

Next deactivate the REQSTB signal (Low), set the Request Bus bit [28] to a "1" and zeros to the rest of the bus for two CLK2X cycles. This will set the LC bit in the System Configuration Register to a "1", and zeros to the remaining bits of the SCR. The rest of the SCR is configured in the last sequence as shown in Table 5.0.

Table 1.0 Power Sequence

Step	Pins / Signals	Procedure	Description	# of CLK2X cycles
1[1]	$\overline{RST}$, V_{MATCH}, V_{DD}, V_{DDQ}, CLK2X, PHASEN	Activate Reset, Ramp up supply & clock signals	$\overline{RST}$ must be active (low) at power up and remain active until Step 3. Power supplies and Clock signals need to ramp up to operating conditions and become stable. VDDQ can not ramp up ahead of VDD.	NA
2	REQSTB, $\overline{RST}$	Reset internal logic and registers	After signals of Step 1 are stable, REQSTB and $\overline{RST}$ must be low for thirty-two CLK2X cycles to insure that all internal logic and registers are fully reset.	32
3	CONFIGIN	Set High	De-activate $\overline{RST}$ signal to commence IPC operations. Set CONFIGIN signal high for a sixteen CLK2X cycles to set the Device ID in the Device Expansion Register	16
4a[2]	REQSTB	Activate	Activate REQSTB (High) for two CLK2X cycles to signal the start of a valid operation.	2
	Command Bus	Select Write Instruction	Using Command Bus select Write instruction, set CMD [3:0] bits to "0100", zeros to rest of bus.	
	Request Bus	Select SCR	Using Request Bus select System Configuration Register, set Request Bus bits [8:1] to "0000 0011", zeros to rest of bus.	
4b[2]	REQSTB	De-activate	De-activate REQSTB (Low).	2
	Request Bus	Set LC bit	Using Request Bus, set the LC bit [28] to a 1 in the SCR, zeros to rest of bus.	
The Background Write of Data Array is described in the next section.				

5331 tbl 01

NOTE:

1. It is very important that the voltage on the input pins never exceeds the VDDQ level by more than 300mV. Higher voltages could turn on the ESD diodes and the device could be exposed to very high electrical currents which would permanently damage the device.

2. If the ASIC/FPGA does not require the handshake signals during initialization then the LC bit does not have to be enabled until the System Configuration Register is configured.

❖ Background Write of Data Array

Overview:

When the IPC device powers up random information is stored in the Data Array. It is recommended that the entire Data Array be initialized to prevent a false match from occurring in an un-initialized location. The Data Array should be initialized to all zeros.

Sequence:

Data Array

The procedure to initialize that Data Array is as follows: Signal the start of the a valid operation by activating the REQSTB signal (High) for two CLK2X cycles. On the Command Bus select the Write instruction, set the CMD bits [3:0] to “0100” and zeros to the rest of the Command Bus. On the Request Bus, select the Access Type set bits [25:24] to “01” for Data Array, set the GMR Select bits [23:22] to "11" for no masking, the Address field bits [15:1] to all zeros (initial Address) and the rest of the Request Bus bits to zeros.

Next deactivate the REQSTB signal (Low), set the Request Bus bits [71:0] to all 0's for two CLK2X cycles. Repeat this sequence for all of the 64K Data entries, increment the Address location by 1 until all of the entries have been initialized.

Table 2.0 Background Sequence

Step	Pins / Buses	Procedure	Description	# of CLK2X cycles
5a(3)	REQSTB	Activate	Activate REQSTB (High) for two CLK2X cycles to signal the start of a valid operation.	2
	Command Bus	Select Write Instruction	Using the Command Bus select Write instruction, set CMD [3:0] bits to "0100", zeros to rest of bus.	
	Request Bus	Select Data Entries	Using Request Bus set: Access Type bits [25:24] to "01" for Data, GMR Select bits [23:22] to "11" for no masking, Address field bits [15:1] to all zeros (initial Address) and zeros to the rest of bus.	
5b(3)	REQSTB	De-activate	De-activate REQSTB (Low).	2
	Request Bus	Write 0's to Data Array	Set "0's" on all Request Bus bits [71:0].	
Repeat Step 5 for every Data Address location.				
The Initialization of the Global Mask Registers is described in the next section.				

5331 tbl 02

NOTE:

3. Repeat this step for the entire Data Array, increment the Address location by 1 until all 64K Data entries are initialized. This takes 262,144 CLK2X cycles.

❖ Initialization of Global Mask Registers (GMRs)

Overview:

The Global Mask Registers (GMRs) do not have any defined initialized state. If the user intends on using the GMRs then the appropriate register(s) must be initialized. If the user does not intend on using the GMRs, no initialization is required. There are a total of 15 Global Mask Registers in the IPC that are used during lookup and write operations. There are seven GMRs assigned for x72 lookup widths, four for x144 Lookup widths and four for x288 Lookup widths. There are three GMRs used for write instructions. All writes are of 72 bits. Each GMR contains 72 bits. Table 3.1 shows the address assignment of each register.

Sequence:

The procedure to initialize the GMRs is as follows: Signal the start of the next valid operation by activating the REQSTB signal (High) for two CLK2X cycles. On the Command bus select the Write instruction, set the CMD bits [3:0] to "0100" and zero's to the rest of the Command Bus. On the Request Bus select the address of the GMR, using Request Bus bits [8:1] and zeros to the rest of the Request Bus bits.

Next deactivate the REQSTB signal (Low), set the configuration of the GMR addressed on Request Bus bits [71:0] for two CLK2X cycles. Repeat this sequence for every GMR that is to be used.

Table 3.1 — GMRs Addresses

Address	Register	Function
0001 0000	Global Mask Register 10	72 bit Lookup / Write
0001 0001	Global Mask Register 11	72 bit Lookup / Write
0001 0010	Global Mask Register 12	72 bit Lookup / Write
0001 0011	Global Mask Register 13	72 bit Lookup
0001 0100	Global Mask Register 14	72 bit Lookup
0001 0101	Global Mask Register 15	72 bit Lookup
0001 0110	Global Mask Register 16	72 bit Lookup
0010 0000	Global Mask Register 20	144 bit Lookup
0010 0001	Global Mask Register 21	144 bit Lookup
0010 0100	Global Mask Register 24	144 bit Lookup
0010 0101	Global Mask Register 25	144 bit Lookup
0011 0000	Global Mask Register 30	288 bit Lookup
0011 0001	Global Mask Register 31	288 bit Lookup
0011 0010	Global Mask Register 32	288 bit Lookup
0011 0011	Global Mask Register 33	288 bit Lookup

5331 tbl 03

Table 3.0 GMR Sequence

Step	Pins / Buses	Procedure	Description	# of CLK2X cycles
6a(4)	REQSTB	Activate	Activate REQSTB (High) for two CLK2X cycles to signal the start of a valid operation.	2
	Command Bus	Select Write Instruction	Using the Command Bus select Write instruction, set CMD [3:0] bits to "0100", zeros to rest of bus.	
	Request Bus	Select GMR	On the Request Bus, set the GMR address on bits [8:1], set zeros on the rest of Request Bus.	
6b(4)	REQSTB	De-activate	De-activate REQSTB (Low).	2
	Request Bus	Configure GMR	Configure the GMR using Request Bus bits [71:0].	
The initializations of the Reply Width Registers is described in the next section.				

5331 tbl 04

NOTE:
4. Repeat this step for each register. To determine the total number of CLK2X cycles needed, add four CLK2X cycles for every register initialized.

❖ Initialization of Reply Width Registers (RWRs)

Overview:

The Reply Width Registers (RWRs) do not have any defined initialized state and must be initialized. There are four Reply Width Registers in the IPC that are used for lookup and write operations. The information stored in the RWRs for the specified width is sent along with the Index after a lookup operation. If the user intends on using this feature then the RWRs should be configured as needed. If the user does not intend on using this feature then the RWRs need to be configure to all zeros. Table 4.1 shows the address assignments of each register.

Sequence:

The procedure to initialize the Reply Width Registers is similar to that of GMRs. Signal the start of the next valid operation by activating the REQSTB signal (High) for two CLK2X cycles. On the Command bus select the Write instruction, set the CMD bits [3:0] to "0100" and zero's to the rest of the Command Bus. On the Request Bus select the address of the RWR, using Request Bus bits [8:1] and zeros to the rest of the Request Bus bits.

Next deactivate the REQSTB signal (Low), set the configuration of the RWR addressed on Request Bus bits [71:0] for two CLK2X cycles. Repeat this sequence for every RWR that is to be used.

Table 4.1 — RWRs Addresses

Address	Register	Function
0000 0100	Reply Width Register 0	Device Operation
0000 0101	Reply Width Register 1	Device Operation
0000 0110	Reply Width Register 2	Device Operation
0000 0111	Reply Width Register 3	Device Operation

5331 tbl 06

Table 4.0 SRR Sequence

Step	Pins / Buses	Procedure	Description	# of CLK2X cycles
7a[4]	REQSTB	Activate	Activate REQSTB (High) for two CLK2X cycles to signal the start of a valid operation.	2
	Command Bus	Select Write Instruction	Using the Command Bus select Write instruction, set CMD [3:0] bits to "0100", zeros to rest of bus.	
	Request Bus	Select RWR	On the Request Bus, set RWR address on bits [8:1], set zeros on the rest of Request Bus.	
7b[4]	REQSTB	De-activate	De-activate REQSTB (Low).	2
	Request Bus	Configure RWR	Configure the RWR using Request Bus bits [71:0].	
The initialization of the System Configuration Register is described in the next section.				

5331 tbl 05

NOTE:

4. Repeat this step for each register. To determine the total number of CLK2X cycles needed, add four CLK2X cycles for every register initialized.

❖ Initialization of System Configuration Register (SCR)

Overview:

The final register that needs to be initialized is the System Configuration Register to enable the device for operation. The Index Bus will remain tri-stated until the Enable bit (EN) is set to a 1 in the System Configuration Register. After this register is configured the device is ready for operation.

Sequence:

The procedure to initialize the System Configuration Register is the same as the GMRs and RWRs. Signal the start of the next valid operation by activating the REQSTB signal (High) for two CLK2X cycles. On the Command bus select the Write instruction, set the CMD bits [3:0] to "0100" and zero's to the rest of the Command Bus. On the Request Bus select the SCR, set the Request Bus bits [8:1] to "0000 0011" and zeros to the rest of the Request Bus bits.

Next deactivate the REQSTB signal (Low) for two CLK2X cycles. Using the Request Bus configure the SCR as follows: Set the Reserved bits [39:32], [27:12] and [4:2] to 0's; Set bits [31:28] to 1's (sets the EN, SR, LS, LC bits), SR=1 means ZBT™ SRAM attached to Index Bus; Set the IPC Grp bits [11:5] to 0's; Set the PD bits [1:0] to "00" , a delay of 0 is set so the RDACK, HITACK and VALID signals are driven with the Index; Set zeros to the rest of the Request Bus bits. Refer to the IDT75T54100 Datasheet for further details on the System Configuration Register.

Table 5.0 SCR Sequence

Step	Pins / Signals	Procedure	Description	# of CLK2X cycles
8a	REQSTB	Activate	Activate REQSTB (High) for two CLK2X cycles to signal the start of a valid operation.	2
	Command Bus	Select Write Instruction	Using the Command Bus select Write instruction, set CMD [3:0] bits to "0100", zeros to rest of bus.	
	Request Bus	Select SCR	Using Request Bus select System Configuration Register, set Request Bus bits [8:1] to "0000 0011", zeros to rest of bus.	
8b(5)	REQSTB	De-activate	De-activate REQSTB (Low).	2
	Request Bus	Configure SCR	Using Request Bus configure SCR, set EN, SR, LS and LC bits [31:28] to 1's and zeros to rest of bus. SR=1 defines ZBT™ SRAM output timing has been selected.	
Commence Normal Operation.				

5331 tbl 07

NOTE:
5. Refer to the Datasheet for more details on the System Configuration Register and Output Timing (with ZBT™ SRAM).

◆ Read Only Registers

The Identification Register, Internal Test Register, Depth Expansion Register and the Search Result Registers are read only registers and cannot be initialized through an IPC write operation. The information stored in the Identification Register and Internal Test Register is encoded in the device during manufacturing. The information (Device ID) stored in the Depth Expansion Register is hardware controlled by the CONFIGIN signal and is programmed in Step 3. The Search Result Registers will be dynamically changing as the device is used. It is important to realize that these registers initialize in a random state and should not be used for an Indirect Write or Read operation until after they have been updated from a previous Lookup operation.

Table 6.0 Single Device Summary Table

Step	Pins / Signals	Procedure	Description	CLK2X cycles
1(1)	$\overline{RST}$, VDD, VDDQ, VMATCH CLK2X, PHASEN,	Activate Reset, Ramp up supply & clock signals	$\overline{RST}$ must be active before powering up the device and remain active until Step 3. Power supplies and Clock signals need to ramp up to operating conditions and become stable. VDDQ can not ramp up ahead of VDD.	NA
2	REQSTB, $\overline{RST}$	Reset internal logic and registers	After signals of Step 1 are stable, REQSTB and $\overline{RST}$ must be low for thirty-two CLK2X cycles to insure that all internal logic and registers are fully reset.	32
3	CONFIGIN	Set High	De-activate $\overline{RST}$ signal to commence IPC operations. Set CONFIGIN signal high for sixteen CLK2X cycles to set the Device ID in the Device Expansion Register.	16
4a(2)	REQSTB	Activate	Activate REQSTB (High) for two CLK2X cycles to signal the start of a valid operation.	2
	Command Bus	Select Write Instruction	Using Command Bus select Write instruction, set CMD [3:0] bits to "0100", zeros to rest of bus.	
	Request Bus	Select SCR	Using Request Bus select SCR, set Request Bus bits [8:1] to "0000 0011", zeros to rest of bus.	
4b(2)	REQSTB	De-activate REQSTB	De-activate REQSTB (Low) .	2
	Request Bus	Set LC bit	Using Request Bus, set the LC bit [28] to a 1 in the SCR, zeros to rest of bus.	
5a(3)	REQSTB	Activate	Activate REQSTB (High) for two CLK2X cycles to signal the start of a valid operation.	2
	Command Bus	Select Write Instruction	Using the Command Bus select Write instruction, set CMD [3:0] bits to "0100", zeros to rest of bus.	
	Request Bus	Select Data Entries	Using Request Bus set: Access Type bits [25:24] to "01" for Data, GMR Select bits [23:22] to "11" for no masking, Address field bits [15:1] to all zeros (initial Address) and zeros to the rest of bus.	
5b(3)	REQSTB	De-activate	De-activate REQSTB (Low).	2
	Request Bus	Write 0's to Data	Set "0's" on all Request Bus bits [71:0].	
Repeat Step 5 for every Data Address location.				
6a(4)	REQSTB	Activate	Activate REQSTB (High) for two CLK2X cycles to signal the start of a valid operation.	2
	Command Bus	Select Write Instruction	Using the Command Bus select Write instruction, set CMD [3:0] bits to "0100", zeros to rest of bus.	
	Request Bus	Select GMR	On the Request Bus set the GMR address bits [8:1], set zeros on the rest of Request Bus.	
6b(4)	REQSTB	De-activate	De-activate REQSTB (Low).	2
	Request Bus	Configure GMR	Configure the GMR using Request Bus bits [71:0].	
7a(4)	REQSTB	Activate	Activate REQSTB (High) for two CLK2X cycles to signal the start of a valid operation.	2
	Command Bus	Select Write Instruction	Using the Command Bus select Write instruction, set CMD [3:0] bits to "0100", zeros to rest of bus.	
	Request Bus	Select RWR	On the Request Bus set the RWR address bits [8:1], set zeros on the rest of Request Bus.	
7b(4)	REQSTB	De-activate	De-activate REQSTB (Low).	2
	Request Bus	Configure RWR	Configure the RWR using Request Bus bits [71:0].	
8a	REQSTB	Activate	Activate REQSTB (High) for two CLK2X cycles to signal the start of a valid operation.	2
	Command Bus	Select Write Instruction	Using the Command Bus select Write instruction, set CMD [3:0] bits to "0100", zeros to rest of bus.	
	Request Bus	Select SCR	Using Request Bus select SCR, set Request Bus bits [8:1] to "0000 0011", zeros to rest of bus.	
8b(5)	REQSTB	De-activate	De-activate REQSTB (Low).	2
	Request Bus	Configure SCR	Using Request Bus configure SCR, set EN, SR, LS and LC bits [31:28] to 1's and zeros to rest of bus. SR=1 defines ZBT™ SRAM output timing has been selected.	
Commence Normal Operation.				

5331 tbl 08

NOTE:

1. It is very important that the voltage on the input pins never exceeds the VDDQ level by more than 300mV. Higher voltages could turn on the ESD diodes and the device could be exposed to very high electrical currents which would permanently damage the device.
2. If the ASIC/FPGA does not require the handshake signals during initialization then the LC bit does not have to be enabled until the SCR is configured.
3. Repeat this step for the entire Data Array, increment the Address location by 1 until all 64K Data entries are initialized. This takes 262,144 CLK2X cycles.
4. Repeat this step for each register. To determine the total number of CLK2X cycles needed, add four CLK2X cycles for every register initialized.
5. Refer to the Datasheet for details on the System Configuration Register description and Output Timing (with ZBT™ SRAM).

❖ Multiple Device Initialization

Introduction

Basic terminology is used to explain the possible options when cascading multiple IPC devices. When describing an IPC system, there can be up to eight IPC's in one system. In a IPC system the device with a Device ID # of 0 is the highest priority device and the first IPC in the system. The device with the highest Device ID # is the last IPC in the system and is the lowest priority device. The lowest priority device in the IPC system acts as the master device, it is the only device that knows if a match is found in any of the higher priority devices. The last (lowest priority) device must have the Last IPC (LC) bit set (High) in its System Configuration Register. Setting the LC bit makes it responsible for driving the RDACK, HITACK and VALID signals for the system to the ASIC/FPGA.

An IPC system can support up to eight distinct IPC groups. An IPC group is defined as either a single IPC device or multiple IPC devices that drive a specific bank of SRAMs. The last device in the IPC group is responsible for driving the SRAM control signals and Index Bus when no operation is ongoing, which prevents the Index Bus from floating. The last (lowest priority) device in the IPC group must have the Last SRAM (LS) bit set (High) in its System Configuration Register to drive the SRAM control signals.

The difference in initializing a multiple device system compared to a single device (Part One) is described in each of the sequences as followed:

◆ Powering up the Multiple Devices

Device Identification:

To set the Device ID of multiple IPC's, the CONFIGIN signal of the highest priority IPC needs to be set high. The CONFIGOUT signal is tied to the CONFIGIN signal of the next highest priority device and so on for the rest of the devices in the system. After the CONFIGIN signals has been active (High) for 16 CLK2X cycles the IPC will to begin to drive the CONFIGOUT signal to the next device in the system. To determine the total number of CLK2X cycles needed to set the Device ID in each IPC device, add 16 CLK2X cycles for each IPC device in the system.

ASIC/FPGA Handshaking of signals:

Only the last (lowest priority) device in the IPC system must have the LC bit set (High) in its System Configuration Register to start the handshaking of signals back to the ASIC/FPGA. All of the other devices in the IPC system can not have the LC bit set.

◆ Background Write of Data Array

It is recommended that the entire Data Array in each of the devices in the IPC system are initialized. If in any of the IPC devices the entire array is not fully initialized to a known state a false match might be realized at an un-initialized location.

◆ Initialization of Global Mask Registers (GMRs)

If there is a need to use GMRs then the user must initialize the appropriate register(s) in each of the IPC devices. If the user does not intend on using the GMRs, no initialization is required.

◆ Initialization of Reply Width Registers (RWRs)

In each of the IPC devices in the IPC system, it is recommended that all four Reply Width Registers be initialized with the Device ID in the three width fields in each of the registers. This useful so the user can identify which IPC device had the match in the IPC system.

◆ Initialization of System Configuration Register (SCR)

The major difference in initializing multiple devices is in the setup of the System Configuration Register in each of the IPC devices. The last (lowest priority) device in the IPC system is responsible for driving the RDACK, HITACK and VALID signals for the system to the ASIC/FPGA and must be the only device to have the Last IPC (LC) bit [28] set in its System Configuration Register.

The other difference is in the setup of the IPC Grp bits. The IPC Grp field bits [11:5] are defined as follows: Bit 5 corresponds to IPC0, bit 6 corresponds to IPC1 and so on through bit 11 corresponding IPC6. To determine an IPC group, only the bits that represent to the higher priority devices in the IPC group must be set high, the bit that represents the lowest priority device in the IPC group must be set low. The bits that represent the other devices within the IPC system but not within the IPC group must also be set low. The lowest priority device recognizes the IPC Grp field bits and the Last SRAM (LS) bit to determine its responsibility to drive the SRAM controls signals and the Index Bus for the IPC group it belongs to.

In defining IPC group the Pipeline Delay (PD) field bits [1:0], IPC Group (IPC Grp) field bits [11:5] and SRAM type (SR) field bit [30] must be set the same in each of the devices in the IPC group. The last (lowest priority) device in the IPC group will be the only device to also have the Last SRAM (LS) bit [29] set in its SCR.

Figure 2.0 Example: Multiple IPC Devices

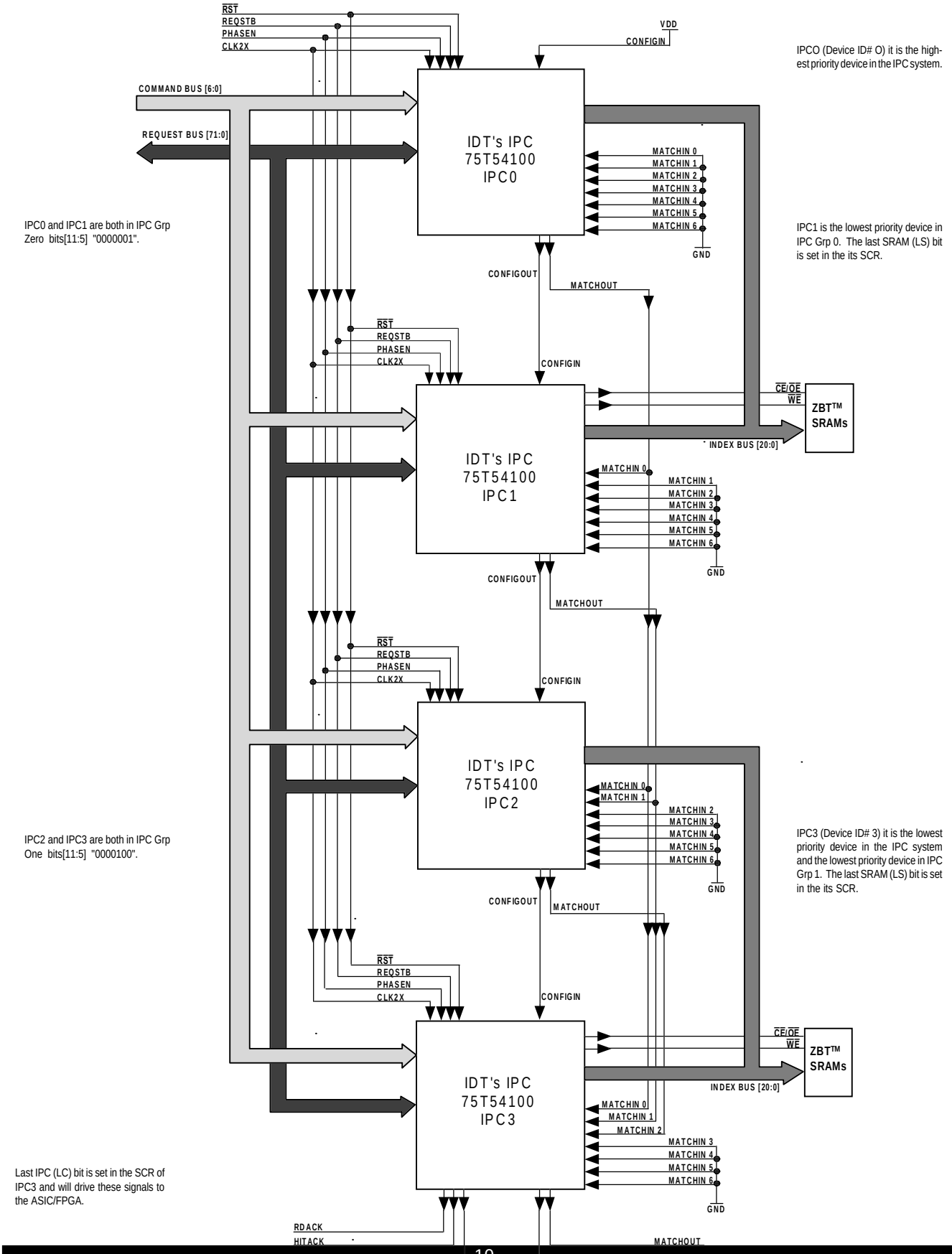


Figure 2.0 Example: Multiple IPC Devices

❖ Example: Multiple IPC Devices

Overview

Figure 2.0 shows the hardware connections for four devices in an IPC system. In the IPC system, IPC0 is the highest priority device and IPC3 is the lowest priority device. IPC3 being the lowest priority device in the IPC system is the only deceive to have the LC (Last IPC) bit set in its System Configuration Register. In this example the IPC system is made up of two IPC groups. IPC0 and IPC1 are part of IPC Grp Zero, IPC2 and IPC3 are part of IPC Grp One. Both IPC1 and IPC3 are the lowest priority device in the their respective IPC Groups therefore they are the only device to have the LS (Last SRAM) bit set in their System Configuration Registers. The difference in initializing this example compared to a single device (Part One) is described in each of the following sequences listed below.

◆ Powering up the Multiple Devices

Device Identification:

The CONFIGIN signal of IPC0 needs to be set high. The CONFIGOUT signal of IPC0 is tied to the CONFIGIN signal of IPC1 and so on for the rest of the devices in the IPC system. After the CONFIGIN signal of IPCO has been high for 16 CLK2X cycles it will to begin to drive the CONFIGOUT signal to the next device in the IPC system. Step 3 of Table 1.0 for this example will take 16x4 CLK2X cycles to set the Device ID in each of the device in the IPC system. Each device must have its Device ID set before moving on to Step 4 of Table 1.0.

ASIC/FPGA Handshaking of signals:

IPC3 being the last IPC needs to have the LC bit set in its System Configuration Register to enable the handshaking of signals back to the ASIC/FPGA. In Step 4a of Table 1.0 when selecting the System Configuration Register, it is also necessary to select IPC3 using the Device ID field of the Request Bus. Set the Device ID field bits [33:26] to “00000011”.

◆ Background Write of Data Array

The procedure to initialize the Data Array in each of the IPC devices is the same except for Device ID. In Step 5a of Table 2.0 when selecting the Data entry, it is also necessary to set the Device ID on the Request Bus. Set Request Bus bits [33:26] to “00000000” to select IPC0, to “00000001” for IPC1, to “00000010” for IPC2 and “00000011” for IPC3.

◆ Initialization of Global Mask Registers (GMRs)

The procedure to initialize the GMRs in each of the IPC devices is the same except for Device ID. In Step 6a of Table 3.0 when selecting the GMR, it is also necessary to set the Device ID on the Request Bus as described previously.

◆ Initialization of Reply Width Registers (RWRs)

It is recommended that the Reply Width Registers in each of the IPC devices be initialized with its Device ID. In Step 7a of Table 4.0 when selecting the RWR, it is also necessary to set the Device ID on the Request Bus as described previously. In Step 7b of Table 4.0 when configuring the RWR set the Device ID of the IPC addressed in the three width fields and zeros to the rest of the Request Bus bits. Refer to the IDT75T54100 Datasheet for further details on the Reply Width Registers.

◆ Initialization of System Configuration Register (SCR)

In this example the System Configuration Register of each of the IPC devices is set up uniquely. To set the SCR of IPC0, IPC1, IPC2 and IPC3 repeat the procedure for each of the devices. In Step 8a of Table 5.0 when selecting the SCR, it is necessary to select the IPC device using the Request Bus bits [33:26] as described previously. In Step 8b of Table 5.0 when configuring the SCR set each IPC up as shown in Table 7.0 below and zeros to the rest of the Request Bus bits.

Table 7.0 — SCR Bit Assignment

IPC Device	Res. [39:32]	EN [31]	SR [30]	LS [29]	LC [28]	Res. [27:12]	IPC Grp [11:5]	Res. [4:2]	PD [1:0]
IPC0	0's	1	1	0	0	0's	0000 001	0's	00
IPC1	0's	1	1	1	0	0's	0000 001	0's	00
IPC2	0's	1	1	0	0	0's	0000 100	0's	00
IPC3	0's	1	1	1	1	0's	0000 100	0's	00

5331 tbl 09

CORPORATE HEADQUARTERS
2975 Stender Way
Santa Clara, CA 95054

for SALES:
831 754-4555
fax: 831-754-4547
www.idt.com

for Tech Support:
ipchelp@idt.com
831 754-4555