Título / Title
DOCC IP DETAILED DESIGN

Redactado por / Written by	**Responsabilidad / Responsibility**	**Fecha / Date**	**Firma / Signature**
Pedro Rodríguez	ASIC Design Engineer		
Revisado por / Reviewed by	**Responsabilidad / Responsibility**	**Fecha / Date**	**Firma / Signature**
Luis Berrojo	ASIC Design Manager		
Aprobado por / Approved by	**Responsabilidad / Responsibility**	**Fecha/ Date**	**Firma / Signature**
Josep Prat	New Products Engineering Manager		
Jaime Arias	P.A. Manager		
Registrado por / Registered by	**Responsabilidad / Responsibility**	**Fecha / Date**	**Firma / Signature**
P. de Francisco	Configuration Control manager		

REGISTRO DE CAMBIOS / CHANGE RECORD			
EDICIÓN ISSUE	**FECHA DATE**	**AUTOR AUTHOR**	**MODIFICACIÓN CHANGE**
01	14/01/02	P. Rodríguez	Document creation

CONTENTS

1 PURPOSE

This document specifies the DOCC IP including scenarios 1 and 2. It includes the following functionalities: OBDH (*On Board Data Handling Bus*) basic modem and DHS (*Data Handling System*) basic modem (for scenery 1), and RTU kernel (for scenery 2).

The resulting DOCC design can work in three different modes:

- "OBDH" compatibility mode. Just to substitute old MA28139 Plessey IC, therefore DOCC is functionally identical to Plessey OBT
- "DHS" mode. It acts as a "4-201 DHS basic modem" compliant CT or RT entity
- Cross-mode, i.e. a "DHS" mode with an IUB interface.

2 DOCUMENTS

2.1 APPLICABLE DOCUMENTS

1. "MA28139 OBDH Bus Terminal", GEC Plessey, Ref: DS3592-4.5, November 1994.
2. "Spacecraft Data Handling Interface Standards", ESA, Ref: TTC-B-01 Issue: 1.
3. "OBDH DIGITAL BUS INTERFACE SPECIFICATION", ESA, Ref. THB/Apo/KZ/1386/av Issue 2
4. "Data Handling System Standard", ESA, Ref: PSS-04-201.
5. "Data Bus Specification", ESA, Ref: PSS-04-255.
6. "DHS Data Bus Modem Specification", ESA, Ref: PSS-04-257.
7. "DOCC ASIC Technical Specification", AEO-003077.
8. "CMOS DOCC ASIC Specification", AEO-003078.
9. "DOCC ASIC Test Plan", AEO-003652.

2.2 REFERENCE DOCUMENTS

1. "VHDL Modelling Guidelines", ESA, Ref: ASIC/001 Issue 1, September 1994.
2. "ASIC Design and Manufacturing Requirements", WDN/PS/700, Issue 2, October 1994.
3. "VHDL Models for Board-level Simulation", WSM/SH/010, Issue 1, February 1996.
4. "PLAN DE DESARROLLO DE ASICs DIGITALES SEMICUSTOM". Ref. AEO/PD61.00-S, edición 01, 21/09/92.
5. "ASIC Design Manual MG2/MG2RT", Ref: TEMIC Semiconductors ATD-TS-GR-R0162, Revision 2.3. July 1997.

2.3 DEFINITIONS AND CONVENTIONS

DOCC: DHS On Board Communication Controller**.**
BAR: Bus Adapter for Reception.
BAT: Bus Adapter for Transmission.
Block Transfer Bus (BT-Bus): An optional component bus of the 4-255 Data Bus, which carries block transmissions between terminals.
CT: Control terminal.
DHS: Data Handling System.
DB: Data Bus.
DBI: Digital Bus Interface.
IP: Intelectual Property.
Interrogation Bus (I-Bus): A mandatory component bus of the 4-255 Data Bus or the OBDH Bus which carries interrogation words from the BC to other bus terminals.
Interrogation Word (Interrogation): Is the 32 data bit sequence, which is transmitted by the BC into the I-Bus to one or more RT in each slot.

IUB: Internal User Bus.
Modem Service (Service): A capability that is provided by the modem.
OBDH: On Board Data Handling.
Response Bus (R-Bus): A mandatory component bus of the 4-255 Data Bus or the OBDH Bus, which carries response words between terminals.
Response Word (Response): Is the data bit sequence transmitted by a RT into the R-Bus in response to an Interrogation from the BC.
RT: Remote terminal.
SAP: Service access point.
Service User (User): Entity that makes use of a service.

3 CAD TOOLS AND DESIGN DATA BASE ORGANIZATION DESCRIPTION

3.1 CAD TOOLS

The tools used in this design are:

- VHDL Simulation: LEAPFROG v. 2.7. by CADENCE
- VHDL Simulation Coverage: Vnavigator by TransEDA
- VHDL Synthesis: SYNOPSYS v. 2001.08.
- VERILOG Simulation: Verilog v. 2.5.12.
- ACTEL Synthesis: ACTmap VHDL synthesis R-2 1998 3.3.0.15.
- ACTEL Place & Route: DESIGNER R-2 1998 3.3.0.15.
- ACTEL Programming: ACTIVATOR 2s.
- Foundry Tools MHS_DK, for synopsys v. 5.1, for Cadence v. 8.1

3.2 DESIGN DATA BASE ORGANIZATION DESCRIPTION

A data CD contains the files related with this document and needed for manufacturing.

The different Files used in the design of the IP have been distributed in four directories as described bellow. All this information is contained in the DOCC directory.

- CODIGO: Here are included the files that contain the VHDL code. These files are separated in two directories:
  - BASIC: Includes the packages of the Basic library.
    - attributes.vhd – standard attributes.
    - std_logic_arith.vhd – arithmetical functions.
    - std_logic_misc.vhd – includes the Xor_Reduce fucntion used in the DOCC design.
  - DESIGN: Includes the VHDL code of the DOCC ASIC design, and also the VHDL files for the test modules and for the test benches.
    - Docc.vhd – VHDL code of the DOCC ASIC design.
    - CLKSelection.vhd – selection of the clocks and resets depending on if the ASIC works in normal or test mode.
    - BUFGS.vhd: The internal clock buffers are substituted by shortcuts in order to simulate the design, before the pads are inserted by Synopsys.
    - GenBAR1.vhd – test module that generates the signals for the BAR1 (in OBDH mode) or the IRS Service (in DHS mode).
    - GenBAR2.vhd – the same as for BAR2 or RRS Service.
    - GenBAR3.vhd – the same as for BAR3 or BTRS Service.

  - GenBAT1.vhd – the same as for BAT1 or ITS/RTS Service
  - GenBAT2.vhd – the same as for BAT2 or BTTS Service.
  - GenRTRTU.vhd – test module that generates the NIDS1n, NIDS2n, RIDS1n and RIDS2n signals that codify the IUB commands to be checked in RT mode.
  - GenCTRTU.vhd – test module that generates the ROR signals that codify the IUB commands in CT mode.
  - tb_OBDHRT.vhd – test bench to prove the ASIC working in OBDH and RT mode.
  - tb_OBDHCT.vhd – test bench for the OBDH in CT mode.
  - tb_OBDHRTRTU.vhd – test bench to probe the RTU Kernel in RT mode (for OBDH and DHS, changing only the configuration signal OBDHpDHSn).
  - tb_OBDHCTRTU.vhd – test bench to test the RTU Kernel in CT mode.
  - tb_DHSRT.vhd – test bench for the DHS in RT mode.
  - tb_DHSCT.vhd – test bench for the DHS in CT mode.
- LEAPFROG: In this directory are the files needed to simulate the VHDL code with the tool called Leapfrog v2.7 by CADENCE.
  - BASIC – directory that contains the files produced in the compilation of the VHDL files included in the CODIGO/BASIC directory.
  - DESIGN – files generated in the compilation of the CODIGO/DESIGN files.
  - sv.shm – data saved of the last simulation.
  - cds.lib – set leapfrog libraries.
  - hdl.var – set leapfrog variables.
  - traceOBDHRT – contains the name of the signals to be traced when testing the DOCC ASIC working in OBDH and RT mode.
  - traceOBDHCT – likewise in OBDH and CT mode.
  - traceOBRTRTU – likewise when testing the RTU kernel in RT mode.
  - traceOBCTRTU – likewise when testing the RTU kernel in CT mode.
  - traceDHSRT – likewise in DHS and RT mode.
  - traceDHSCT – likewise in DHS and CT mode.
- VNAVIGATOR: Contains all the files generated by the tool VNavigator by TransEda. This tool is used to make an estimation of the code quality.
- SYNOPSYS: Contains the SYNOPSYS files, including database netlists (synthesis results), synthesis scripts and reports. The following files appear:
  - DOCC.db: Synthesised netlist in SYNOPSYS database format for MG2RT.
  - DOCCSCAN.db: Synthesised netlist, including inserted scan chains, in SYNOPSYS database format.
  - DOCCATC.db: Synthesised netlist in SYNOPSYS database format for ACTEL.
  - DOCC1.vdb: Test pattern file in SYNOPSYS database format. First file.
  - DOCC2.vdb: Test pattern file in SYNOPSYS database format. Second file.
  - DOCC3.vdb: Test pattern file in SYNOPSYS database format. Third file.
  - DOCC4.vdb: Test pattern file in SYNOPSYS database format. Fourth file.

There are the following subdirectories:

- BASIC: analysis result files of the BASIC library.
- DESIGN: analysis result files of the DESIGN library (Docc and clockSelection modules).
- SCR: Contains Synthesis scripts:
  - Docc.scr for synthesising the design with MG2RT.
  - DoccScan.scr: inserts scan chains and performs ATPG.
  - DoccActel.scr: synthesis by using ACTEl as target technology.
- REPORT: contains different reports for area, timing, scan chains, and fault coverage for the synthesis with MG2RT.
- REPORTACT: contains different reports for area, timing, etc, for the synthesis with ACTEL.

4 INPUT/OUTPUT SIGNALS

4.1 I/O FUNCTIONAL DESCRIPTION

4.1.1 Control Signals

Control signals	
Signal	**Type**
LOSCEXTpINTn	I
OBDHpDHSn	I
CTpRTn	I
RESETN	I
SIMUL	I
CASCADING	I
LOSCINT	I
LOSCEXT	I
LOSCDIV4	O
TIMEOUTn	O

Table 4.1 Control signals for both standards.

- **LOSCEXTpINTn:** Input signal.

Select between the internal ASIC's oscillator and an external clock.

If LOSCEXTpINTn = '1', the Asic's clock is proportioned externally through the input LOSCEXT.

If LOSCEXTpINTn = '0', the Asic's clock is generated by an internal oscillator.

- **OBDHpDHSn:** Input signal.

Select between "OBDH" standard and "DHS" standard:

If OBDHpDHSn = '1', selects OBDH standard

If OBDHpDHSn = '0', selects DHS standard

- **CTpRTn:** Input signal.

Select between CT mode and RT mode:

If CTpRTn = '1', the modem is in CT ("*Control Terminal*") mode

If CTpRTn = '0', the modem is in non controller mode (RT, "*Remote Terminal*")

- **RESETN:** Input signal (Schmidt).

Reset the device when low.

- **SIMUL:** Input signal.

Controls simultaneous transmission through nominal and redundant response buses. This signal controls both DHS and OBDH modems and CT and RT modes.

- C**ASCADING:** Input signal.

When asserted at low level, normal bus selection matrix. If asserted at high level, both buses deselected. This signal controls both DHS and OBDH modems and CT and RT modes.

- **LOSCINT:** Input signal (Schmidt).

Is included in order to add an internal RC oscillator, when the IP is implemented as ASIC. I.e. Two sources can be used for the internal net, an Input pad as well as a RC internal oscillator. A control line performs selection between the two choices. It's an 8 MHz clock signal.

- **LOSCEXT:** Input signal (Schmidt).

Local oscillator input signal. It's multiplexed with an internal RC oscillator. I.e. Two sources can be used for the internal net, an Input pad as well as a RC internal oscillator. A control line performs selection between the two choices. It's an 8 MHz clock signal.

- **LOSCDIV4:** Output signal.

This is a clock which frequency is four times less than the LOSC frequency. It can be used as master clock for the DHS mode by connecting the LOSCDIV4 pin with the MASTEROSCIN pin.

- **TIMEOUTn:** Output signal.

Low when I-Bus timeout. Indicates the period when no bus is available to transmission, due no activation. Also indicates that the bus selected has changed.

4.1.2 Data Bus Signals

Buses signals (both standards)	
Signal Name	**Type**
NIDS1n	I
NIDS2n	I
RIDS1n	I
RIDS2n	I
NRDS1n	I
NRDS2n	I
RRDS1n	I
RRDS2n	I
RR1n	O
RR2	O
RR3n	O
RR4	O
NRE	O
RRE	O
NBDS1n	I
NBDS2n	I
RBDS1n	I
RBDS2n	I
BR1n	O
BR2	O
BR3n	O
BR4	O
NBE	O
RBE	O

Table 4.2 Modem signals toward/from discriminators and drivers

- **NIDS1n, NIDS2n**: Input Signals.

Signals recovered from Nominal I-Bus in RT.

- **RIDS1n, RIDS2n:** Input Signals.

Signals recovered from Redundant I-Bus in RT. It could be used in CT mode for diagnostics.

- **NRDS1n, NRDS2n:** Input signals.

Signals recovered from Nominal R-Bus in CT. Also used in RT "Listening mode".

- RRDS1n, RRDS2n: Input signals.

Signals recovered from Redundant R-Bus in CT. Also used in RT "Listening mode".

- **NBDS1n, NBDS2n:** Input signals.

Signals recovered from Nominal BT-Bus in CT mode. Also used in RT mode for bi-directional BTB.

- **RBDS1n, RBDS2n:** Input signals.

Signals recovered from Redundant BT-Bus in CT mode. Also used in RT mode for bi-directional BTB.

- RR1n, RR2, RR3n, and RR4: Output signals.

They come from the sampled data to modem and it is targeted to R-Bus (RT mode) or I-Bus (CT mode) since both modes never exist simultaneously.

- **NRE, RRE:** Output signals.

These signals control transmission to the R-Bus through nominal or redundant Buses.

- BR1n, BR2, BR3n, and BR4: Output signals.

They come from the sampled data fed to modem and it is targeted to BT-Bus in both RT and CT modes.

- **NBE, RBE:** Output signals.

These signals control transmission to the BT-Bus through nominal or redundant Buses.

4.1.3 "DHS" Physical Level SAP's Signals

"DHS" standard		
SAP	**Signal**	**Type**
BTS	BTSCLKOUT	O
BTS	BTSSYNC	O
BTS	BTSENABLE	I
BTS	MASTEROSCIN	I
BSS	ACTIVEN	O
BSS	ACTIVER	O
BSS	SELECTOUTN	O
BSS	SELECTOUTR	O
BSS	SELECTINN	I
BSS	SELECTINR	I
ITS	ITSENABLE	I
ITS	ITSDATAIN	I
ITS	ITSCLOCKOUT	O
IRS	IRSENABLE	I
IRS	IRSDATAOUT	O
IRS	IRSVALIDOUT	O
IRS	IRSCLOCKOUT	O
RTS	RTSENABLE	I
RTS	RTSDATAIN	I
RTS	RTSCLOCKOUT	O
RRS	RRSENABLE	I
RRS	RRSDATAOUT	O
RRS	RRSVALIDOUT	O
RRS	RRSCLOCKOUT	O
BTTS	BTTSENABLE	I
BTTS	BTTSDATAIN	I
BTTS	BTTSCLOCKOUT	O
BTRS	BTRSENABLE	I
BTRS	BTRSDATAOUT	O
BTRS	BTRSVALIDOUT	O
BTRS	BTRSCLOCKOUT	O

Table 4.3 Modem signals in "DHS" standard

4.1.3.1 BTS Service

- **MASTEROSCIN**: Input signal.

Only for CT modems. It is a train of pulses of frequency $f_{Master\_Osc\_In}$. It is used to provide the BTS a reference oscillator input. $f_{Master\_Osc\_In}$ is four times BTSCLKOUT's frequency.

- **BTSCLKOUT**: Output signal.

Is a rectangular pulse train, the interval between two consecutive falling edges mark the Data Bus bit period. It has a frequency f_{bit} ($=1/t_C$) and a mark-space ratio of MSR_{bit}. In disabled mode it is maintained at a steady high level. It is derived internally from MASTEROSCIN.

- **BTSSYNC**: Output signal.

Each rising edge signifies that the next bit period corresponds to bit '0' of a bus slot. This signal is high during bus bit period 31. In disable mode it is maintained at steady low level.

- **BTSENABLE (H)**: Input signal.

Only for CT modems. In high state the Bus Timing Service is enabled otherwise it is disabled.

4.1.3.2 BSS Service

- **ACTIVEN (H)**: Output signal.

It indicates that the associated bus (nominal) is active or inactive, that is, the activity criteria is checked and signal is updated accordingly.

- **SELECTOUTN (H)**: Output signal.

It indicates that the associated bus is currently selected.

- **SELECTINN (H)**: Input signal.

A request for the nominal set of buses to be used when asserted. This signal exists only on CT modems.

- **ACTIVER (H):** Output signal.

It indicates that the associated bus (redundant) is active or inactive, that is, the activity criteria is checked and signal is updated accordingly.

- **SELECTOUTR (H)**: Output signal.

It indicates that the redundant bus is currently selected.

- **SELECTINR (H)**: Input signal.

A request for the redundant set of buses to be used when asserted. This signal exists only on CT modems.

4.1.3.3 ITS Service

- **ITSENABLE (H)**: Input signal.

When is asserted by the user and the BTS is enabled, the ITS must operate in enabled mode. If the signal is not asserted then the ITS must operate in disabled mode.

- **ITSDATAIN**: Input signal.

Must be used by the user to provide the data. The ITS must sample it on each falling edge of the ITSCLOCKOUT.

- **ITSCLOCKOUT**: Output signal.

In enabled mode the signal shall convey a train of negative going rectangular pulses. The falling edge of each pulse is the data sampling instant for the ITSDATAIN to be modulated onto the driver signals. It must be the same as BTSCLKOUT in CT modems. In disabled mode must be maintained at a steady high level.

4.1.3.4 IRS Service

- **IRSENABLE (H)**: Input signal.

When is asserted by the user, and the BTS is in enabled mode IRS must operate in enabled mode. When is not asserted IRS must operate in disabled mode.

- **IRSDATAOUT**: Output signal.

Must provide the received data to the user. In disabled mode must be maintained at steady low level.

- **IRSVALIDOUT (H)**: Output signal.

Must indicate the validity of the modulated symbol received in each bit. In disabled mode must be maintained at steady low level.

- **IRSCLOCKOUT**: Output signal.

In enabled mode must be a train of low going pulses during the receiving of data from the I-Bus. The falling of each pulse is the data sampling instant for the IRSDATAOUT and IRSVALIDOUT (H) that indicates the user that data can be read form the SAP. In disabled mode must be maintained at a steady high level.

4.1.3.5 RTS Service

- **RTSENABLE (H)**: Input signal.

When is asserted by the user, and the BTS is in enabled mode, RTS must operate in enabled mode. When is not asserted RTS must operate in disabled mode. Changes from mode to mode (enabled, disabled) are synchronised with BTSSYNC.

- **RTSDATAIN**: Input signal.

Must be used by the user to provide data to be transmitted on the R-Bus.

- **RTSCLOCKOUT**: Output signal.

In enabled mode must be a train of low going pulses. The falling of each pulse is the data sampling instant for the RTSDATAIN to be modulated onto the driver signals. In disabled mode shall be maintained at a steady high level.

4.1.3.6 RRS Service

- **RRSENABLE (H)**: Input signal.

When is asserted by the user, and the BTS is in enabled mode RRS must operate in enabled mode. When is not asserted RRS must operate in disabled mode.

- **RRSDATAOUT**: Output signal.

Must provide the received data to the user. In disabled mode must be maintained at a steady low level.

- **RRSVALIDOUT (H)**: Output signal.

Must indicate the validity of the modulated symbol received in each bit. In disabled mode must be maintained at a steady low level.

- **RRSCLOCKOUT**: Output signal.

In enabled mode must be a train of low going pulses during the receiving of data from the R-Bus. The falling of each pulse is the data sampling instant for the RRSDATAOUT and RRSVALIDOUT (H) that indicates the user that data can be read from the SAP. In disabled mode or during the period between response transmission burst must be maintained at steady high level.

4.1.3.7 BTTS Service

- **BTTSENABLE (H)**: Input signal.

When is asserted by the user and the BTS is enabled, the BTTS must operate in enabled mode. If the signal is not asserted then the BTTS must operate in disabled mode.

- **BTTSDATAIN**: Input signal.

Must be used by the user to provide the data. The BTTS must sample it on each falling edge of the BTTSCLOCKOUT.

- **BTTSCLOCKOUT**: Output signal.

In enabled mode the signal must convey a train of negative going rectangular pulses. The falling edge of each pulse is the data sampling instant for the BTTSDATAIN. In disabled mode must be maintained at a steady high level.

4.1.3.8 BTRS Service

- **BTRSENABLE (H)**: Input signal.

When is asserted by the user, and the BTS is in enabled mode, BTRS must operate in enabled mode. When is not asserted BTRS must operate in disabled mode.

- **BTRSDATAOUT**: Output signal.

Must provide the received data to the user.

- **BTRSVALIDOUT (H)**: Output signal.

Must indicate the validity of the modulated symbol received in each bit. In disabled mode must be maintained at a steady low level.

- **BTRSCLOCKOUT**: Output signal.

In enabled mode must be a train of low going pulses during the receiving of data from the block transfer bus. The falling of each pulse is the data sampling instant for the BTRSDATAOUT and BTRSVALIDOUT (H) that indicates the user than data can be read from the SAP. In disabled mode or if no data is being received must be maintained at a steady high level.

4.1.4 "OBDH" DBI Signals

"OBDH" standard		
Port	**Signal**	**Type**
CIT	CITMOD	I
CIT	CITSEL	I
CIT	CITDATA	I
CIT	CITCLK	I
CIT	CITSYNC	I
RRR/CRR	RRR/CRRINIT	I
RRR/CRR	RRR/CRRDATA	O
RRR/CRR	RRR/CRRVAL	O
RRR/CRR	RRR/CRRCLK	O
RBR/CBR	RBR/CBRINIT	I
RBR/CBR	RBR/CBRDATA	O
RBR/CBR	RBR/CBRVAL	O
RBR/CBR	RBR/CBRCLK	O
RIR	CTRIRSYNC	I
RIR	CTRIRDATA	I
RIR	CTRIRVAL	I
RIR	CTRIRCLK	I
RIR	RIRSYNC	O
RIR	RIRDATA	O
RIR	RIRVAL	O
RIR	RIRCLK	O
RRT	RRTDATA	I
RRT	RRTEN	I
RBT/CBT	RBT/CBTEN	I
RBT/CBT	RBT/CBTDATA	I

Table 4.4 Modem signals in "OBDH" standard.

4.1.4.1 RIR Output Signals

These signals are active when the RTU Kernel is coupled to the basic modem. It occurs when it operates in RT mode.

- **RIRCLK:** Output signal.

It's a train of pulses of frequency equal to data rate (524488 Hz). One data bit period is equal to one RIRCLK period; first half period low, second half high. It is recovered from I-Bus.

- **RIRSYNC:** Output signal.

It's a pulse of a one bit period duration, at bit 2 of the frame.

- **RIRDATA:** Output signal.

NRZ data received from I-Bus at the falling edge of RIRCLK. Each bit is contained in one data bit period.

- **RIRVAL:** Output signal.

Asserted at the falling edge of RIRSYNC, de-asserted at the RIRCLK falling edge of an invalid RIRDATA bit sampling.

4.1.4.2 RIR Input Signals

These signals become active when the RTU Kernel is isolated to the basic modem. It occurs when the basic modem operates in CT mode.

- **CTRIRCLK:** Input signal.

It's a train of pulses of frequency equal to data rate (524488 Hz). One data bit period is equal to one CTRIRCLK period; first half period low, second half high. It is generated outside the design.

- **CTRIRSYNC:** Input signal.

It's a pulse of a one bit period duration, at bit 2 of the frame.

- **CTRIRDATA:** Input signal.

NRZ data received from I-Bus at the falling edge of CTRIRCLK. Each bit is contained in one data bit period.

- **CTRIRVAL:** Input signal.

Asserted at the falling edge of CTRIRSYNC, de-asserted at the CTRIRCLK falling edge of an invalid CTRIRDATA bit sampling.

4.1.4.3 RRR/CRR Signals

- **CRR/RRRCLK:** Output signal.

A train of pulses of frequency equal to data rate (524488 Hz) One data bit period is equal to one CRR/RRRCLK period; First half period low, second half high. It is recovered from R-Bus, so when no signal is detected at the R-Bus CRR/RRRCLK is at low state.

- **CRR/RRRDATA:** Output signal.

NRZ data recovered from the R-Bus at the falling edge of CRR/RRRCLK. Each bit is contained in one data bit period.

- **CRR/RRRINIT:** Input signal.

A pulse of one bit period during bit '1'. At the CRR/RRRINIT rising edge the CRR/RRRCLK and CRR/RRRDATA signals are set to low level and CRR/RRRVAL signal is asserted.

- **CRR/RRRVAL:** Output signal.

Asserted at the rising edge of CRR/RRRINIT, de-asserted at the CRR/RRRCLK falling edge of a invalid CRR/RRRDATA bit sampling.

4.1.4.4 RBR/CBR Signals

- **CBRCLK:** Output signal.

It's a train of pulses of frequency equal to data rate (524488 Hz). One data bit period is equal to one CBRCLK period; first half period low, second half high. It is recovered from BT-Bus, so when no signal is detected at the BT-Bus CBRCLK is at low state.

- **CBRDATA:** Output signal.

NRZ data recovered from the BT-Bus at the falling edge of CBRCLK. Each bit is contained in one data bit period.

- **CBRINIT:** Input signal.

A pulse of one bit period at bit '1'. At the CBRINIT rising edge the CBRCLK and CBRDATA signals are set to low level and CBRVAL is asserted.

- **CBRVAL:** Output signal.

Asserted at the rising edge of CBRINIT, de-asserted at the CBRCLK falling edge of an invalid CBRDATA bit sampling.

4.1.4.5 CIT Signals

- **CITMOD:** Input signal.

A train of pulses of frequency equal to double of data rate. One data bit period is equal to two CITMOD periods; First half period low, second half period high, each edge marks the limits of pulses for the bus driver signals (RR1n, RR2...).

- **CITCLK:** Input signal.

A train of pulses of frequency equal to data rate. One data bit period is equal to one CITCLK period; first half period low, second half high.

- **CITDATA:** Input signal.

NRZ data at a rate equal to CITCLK bps from the user. Each data is contained in one bit period (time between consecutive CITCLK falling edges) and is sampled in each CITCLK rising edge to be modulated onto I-Bus.

- **CITSYNC:** Input signal.

Is a pulse of one bit period duration at bit '0'. Marks the start of an interrogation.

- **CITSEL:** Input signal.

Assert or de-assert with the falling edge of CITCLK. When is de-asserted (low level), it selects the Nominal Bus. When it is asserted (high level), selects the Redundant Bus.

4.1.4.6 RRT/CRT Signals

- **RRTEN:** Input signal.

Asserted at the falling edge of the RIRSYNC, de-asserted at the RIRCLK falling edge that indicates the end of the last data bit of the transmitted respond word (bit 12 or 20). Enabled response transmission.

- **RRTDATA:** Input signal.

NRZ data at a rate equal to RIRCLK BPS from the user. It begins with the rising edge of the RRTEN. Each data is contained in one bit period (time between consecutive RIRCLK falling edges) and is sampled in each RIRCLK rising edge to be modulated onto R-Bus.

4.1.4.7 RBT/CBT Signals

- **RBTEN:** Input signal.

Asserted at a RIRSYNC falling edge, de-asserted at the RIRCLK falling edge that indicates the end of the last data bit of the transmitted transfer block data.

- **RBTDATA:** Input signal.

NRZ data at a rate equal to RIRCLK bps. It begins with the rising edge of the RBTEN. Each data is contained in one bit period (i.e. time between consecutive RIRCLK falling edges) and is sampled in each RIRCLK rising edge.

4.1.5 IUB Signals

Signal	Type	Description
IUB SECTION		
IRCLK	O	Interrogation rate clock
CTCLK	O	Continuous clock
TRCLK	O	Transfer clock
MLDATA	O	Memory load data to user
MLADD(0:4)	O	Memory load address MSB:LSB (Interrogation bit 10:14)
CHADD(0:7)	O	Channel address 0:7 (Interrogation bit 23:30)
MOSC	O	Mode short command (Interrogation mode bits 19/22 = '1')
MOLC	O	Switch closure ON/OFF command
MOHL	O	High power ON/OFF command
MOBT	O	Mode block Transfer
MODBL	O	Digital bi-level data acquisition
MODS16	O	16 bit serial digital data acquisition
MODS8	O	8 bit serial digital data acquisition
MOANS	O	Single ended analogue data acquisition
MOAND	O	Double ended analogue data acquisition
PC	O	Power on to analogue to digital converter
ANCLK	O	ADC shift clock
SOC	O	Start of conversion
SH	O	Sample/Hold
DIGIN	I	Digital serial data input
ANSIN	I	Analogue serial data input
BCP(1:4)	O	Broadcast pulse (1:4)
BCPVAL	O	Broadcast pulse valid
DATARRT	O	Data to RRT when used as RTU kernel
ENRRT	O	Enable RRT when used as RTU kernel
TA(0:5)	I	Terminal address bit (0:5)
EXTFMT	I	Extended format Enable
EXTMLA1	I	Extended Memory Load Address 1 Enable
EXTMLA2	I	Extended Memory Load Address 2 Enable
OOINH	I	ON/OFF Inhibit of MOLC commands
TAV	I	Terminal available (take low to disable IUB interface)

Table 4.5 IUB signals.

- **IRCLK:** Output signal.

Interrogation rate clock.

- **CTCLK:** Output signal.

Continuous clock ½ RIRCLK.

- **TRCLK:** Output signal.

Transfer clock.

- **MLDATA:** Output signal.

Memory load data to user.

- **MLADD(0:4):** Output signal.

Memory load address MSB: LSB (Interrogation bit 10:14).

- **CHADD(0:7):** Output signal.

Channel address 0:7 (Interrogation bit 23:30).

- **MOSC:** Output signal.

Mode short command (Interrogation mode bits 19/22 = '1').

- **MOLC:** Output signal.

Switch closure ON/OFF command.

- **MOHL:** Output signal.

High power ON/OFF command.

- **MOBT:** Output signal.

Mode block Transfer.

- **MODBL:** Output signal.

Digital bi-level data acquisition.

- **MODS16:** Output signal.

16 bit serial digital data acquisition.

- **MODS8:** Output signal.

8 bit serial digital data acquisition.

- **MOANS:** Output signal.

Single ended analogue data acquisition.

- **MOAND:** Output signal.

Double ended analogue data acquisition.

- **PC:** Output signal.

Power on to analogue to digital converter.

- **ANCLK:** Output signal.

ADC shift clock.

- **SOC:** Output signal.

Start of conversion control for external ADC.

- **SH:** Output signal.

Sample/Hold control for external ADC.

- **DIGIN:** Input signal.

Digital serial data input.

- **ANSIN:** Input signal.

Analogue serial data input.

- **BCP(1:4) :** Output signal.

Broadcast pulse (1:4).

- **BCPVAL:** Output signal.

Broadcast pulse valid.

- **DATARRT:** Output signal.

Data to RRT when used as RTU kernel.

- **ENRRT:** Output signal.

Enable RRT when used as RTU kernel.

- **TA(0:5) :** Input signal.

Terminal address bit (0:5).

- **EXTFMT:** Input signal.

Extended format Enable.

- **EXTMLA1:** Input signal.

Extended Memory Load Address 1 Enable.

- **EXTMLA2:** Input signal.

Extended Memory Load Address 2 Enable.

- **OOINH:** Input signal.

ON/OFF Inhibit of MOLC commands.

- **TAV:** Input signal.

Terminal available (take low to disable IUB interface).

4.1.6 Test Signals

Test Signals	
Signal	**Type**
TESTMODE	I
TESTIBUSCLK	I
TESTRBUSCLK	I
TESTBTBUSCLK	I

Table 4.6 Test Signals.

- **TESTMODE:** Input signal.

Selects between test or normal mode of function.

If TESTMODE = ‘1’, the Asic works in test mode.
If TESTMODE = ‘0’, the Asic works in normal mode.

- **TESTIBUSCLK:** Input signal.

Trough this pin an IbusClk equivalent signal is inyected, when the Asic runs in test mode.

- **TESTRBUSCLK:** Input signal.

Trough this pin a RbusClk equivalent signal is inyected, when the Asic runs in test mode.

- **TESTBTBUSCLK:** Input signal.

Trough this pin a BTbusClk equivalent signal is inyected, when the Asic runs in test mode.

4.1.7 I/O Summary

MODULE	Inputs	Outputs	Total
Control	8	2	10
DataBuses	12	12	24
DHS	13	18	31
OBDH	15	10	25
IUB	13	37	50
Test	4	--	4
Total	65	79	144

Table 4.7 I/O sumary.

Since the ASIC can not be OBDH and DHS at the same time and it will not work in CT and RT mode, many of this inputs and outputs never are going to have stimuli at the same moment. So the design has been made sharing ports between some of those inputs or outputs, which have a similar functionality in different modes. In the following table are shown the signals that will be connected to the same pin.

Inputs and outputs connected with the same pin
RIRDATA = IRSDATAOUT
RIRVAL = IRSVALIDOUT
CRRINIT = RRSENABLE
CRRCLK = RRSCLOCKOUT
CRRDATA = RRSDATAOUT
CRRVAL = RRSVALIDOUT
CBRINIT = BTRSENABLE
CBRCLK = BTRSCLOCKOUT
CBRDATA = BTRSDATAOUT
CBRVAL = BTRSVALIDOUT
CITMOD = MASTEROSCIN
CITDATA = RRTDATA = ITSDATAIN = RTSDATAIN
CITSEL = SELECTINN
RRTEN = ITSENABLE = RTSENABLE
ITSCLOCKOUT = RTSCLOCKOUT
RBTDATA = BTTSDATAIN
RBTEN = BTTSENABLE

Table 4.8 Inputs and Outputs connected with the same pin.

Taking into account the last consideration, the final number of I/O in this design is:

Total I/O Number = 144 – 20 = 124

5 FUNCTIONAL DESCRIPTION

The DOCC IP implementation described in this document follows the functionality described in documents [AD_1] for the OBDH bus, [AD_7] for the DHS bus, and [AD_1] for the RTU module. It also follows the technical specification described in [AD_7]. The following differences are going to be implemented in the DOCC IP against the functionality implemented by Plessey:

1.- The Watchdog and Bus selection module follows the [AD_7] DHS bus implementation. This is in order to standardise the bus selection in a common functionality for both buses.

2.- RIR signals are divided in input signals and output signals, while they are bi-directional signals in the MA28139 ASIC. In that way some technological problems are avoided.

3.- VAL signals timing is delayed ¼ bit. Thus, waveforms on pages 7 and 8 in AD_1 changes slightly. This is applicable to RIRVAL, CRRVAL, and CBRVAL signals on ports BAR 1, 2, and 3. The reason for this change is to perform a synchronous design that otherwise will be impossible. The signals RIRSYNC, RRE, NRE, RBE and NBE are also delayed ¼ bit due the same reason (pages 5,6,7 in AD_1).

4.- The signal CITINV has been eliminated.

5.- SHORTRESP and its function have not been implemented.

6.- There is a new signal, CASCADING, that is used to deselect all the buses. It works for OBDH and DHS and in CT and RT modes.

7.- For OBDH the inverted sync pattern LOLOLOHOHOHO is also valid, as for DHS.

8.- All the references to the Bus Monitor have been avoided in this document. The reason is that in terms of VHDL code, RT and BM are the same design. So the RT mode refers to both types of equipment. The pin RTpBMn is not necessary.

6 DETAILED DESCRIPTION

6.1 BLOCK DIAGRAM

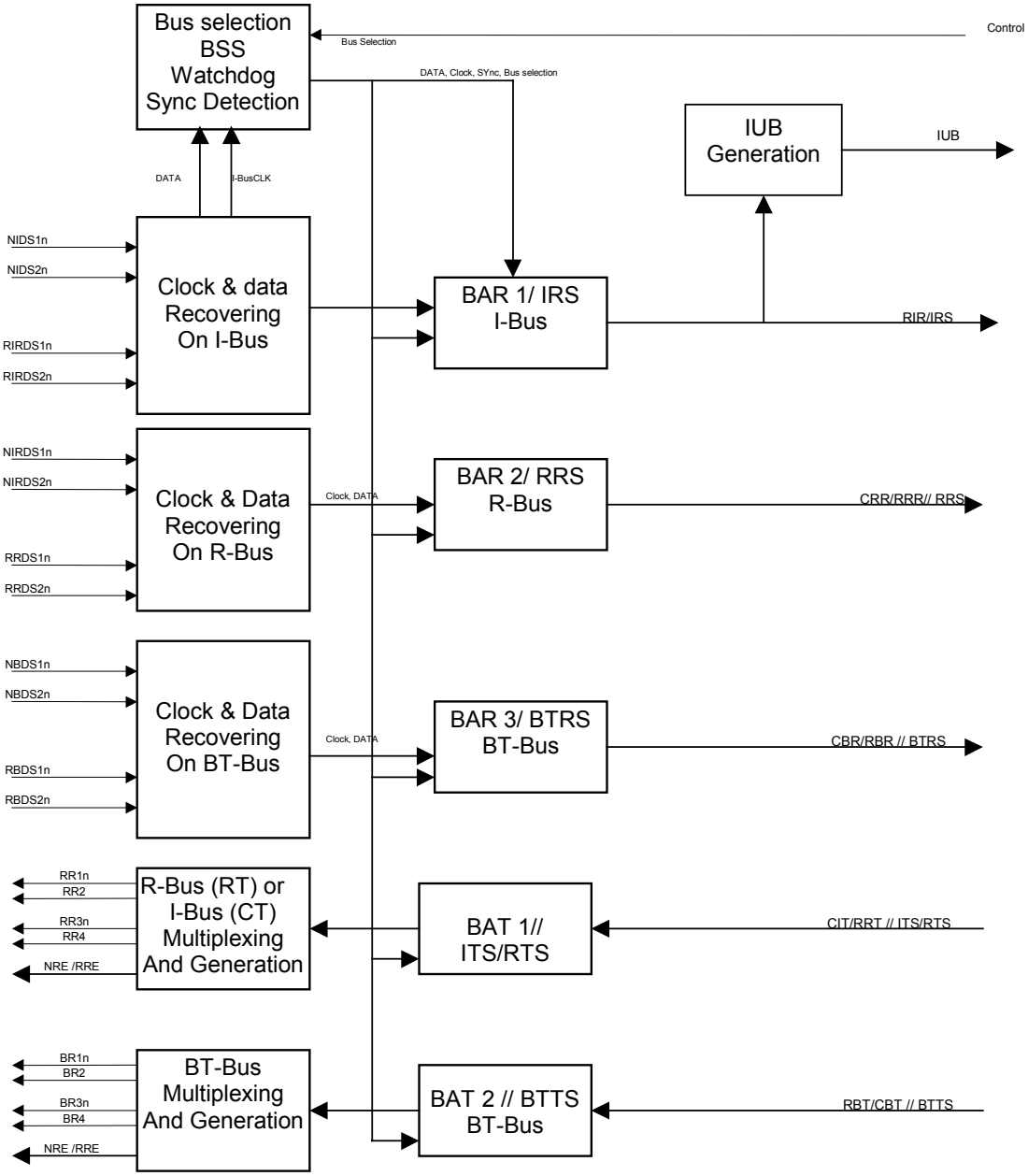


Figure 6.1 DOCC Block Diagram.

6.2 INTERFACE WITH THE I-BUS

6.2.1 Clock and Data Recovery

CLOCK Signals
- LOSC

Input Signals
- NIDS1n
- NIDS2n
- RIDS1n
- RIDS2n

Output Signals
- IBusNomCLK
- IBusRedCLK
- IBusNomDataReg
- IbusRedDataReg

The digital data coming from CT or RT have to be modulated before be placed in the physical wires of the bus. Also the data flowing trough the data bus to the terminals have to be demodulated into digital signals to be understand by the design. The data and clock arrive to the DOCC design from the I-bus modulated using a Litton code. There are two input signals (NIDS1n and NIDS2n) for the nominal bus, and two for the redundant (RIDS1n and RIDS2n). In Litton modulation a unique scheme is assigned to logical '0' and '1'. A modulation clock with a frequency four times of the data clock has to be used, since four changes in Litton signal happen in each bit time. Litton Patterns have three possible voltage values: 'H' positive voltage, 'L' negative voltage or 'O' zero.

In the following graphics a '1'nd a '0' transmission are illustrated:

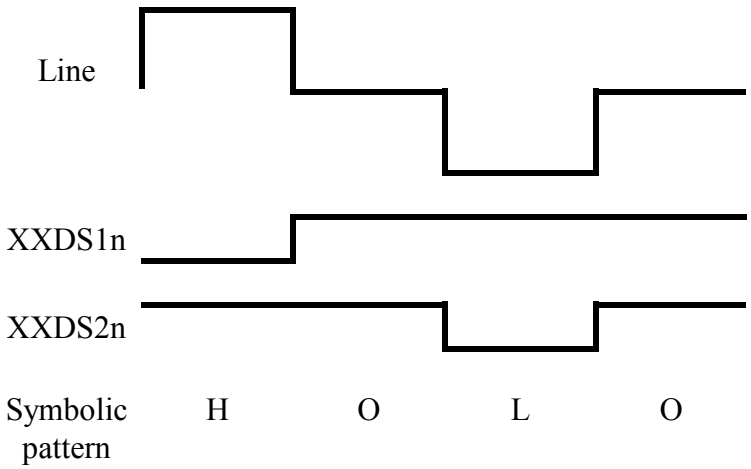


Figure 6.2 Logical '1'.

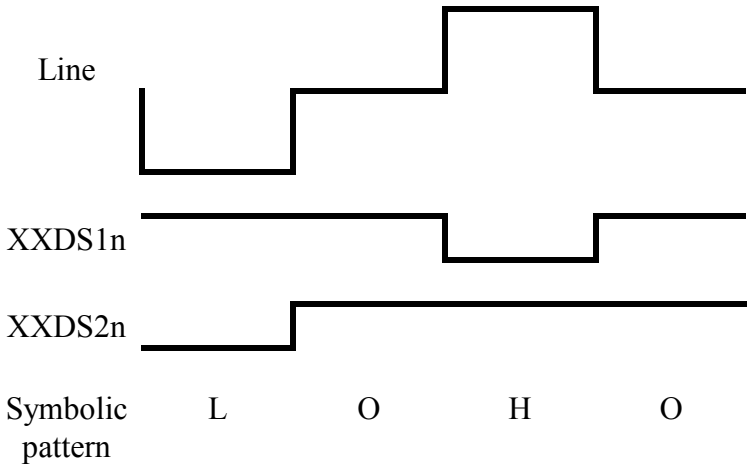


Figure 6.3 Logical '0'.

Also a synchronisation pattern is send at the beginning of every slot transmission. It includes an invalid '1' and '0' codification that make easy the synchronisation header recognition and appears in the following figure:

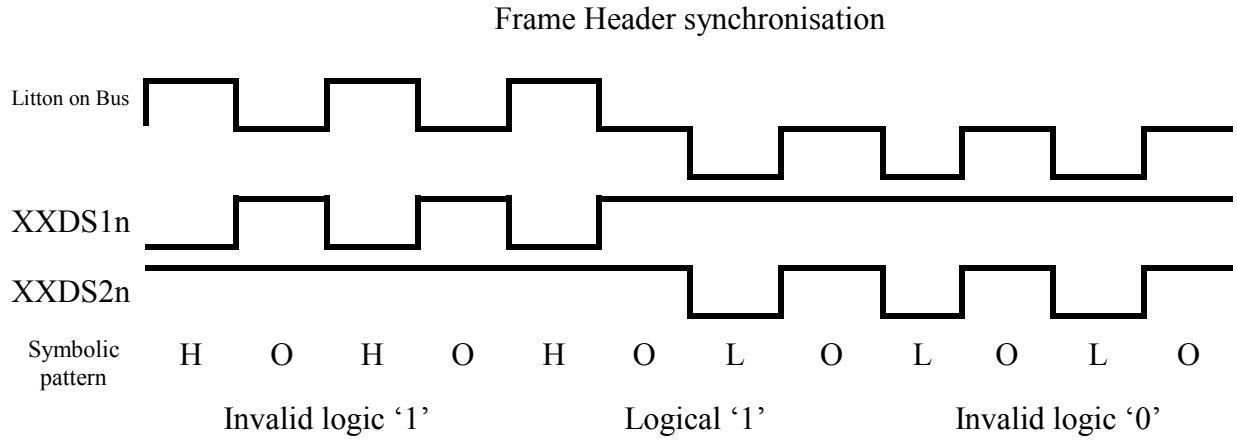


Figure 6.4 Data bus Synchronisation pattern (4-255 or "OBDH").

It must be noted that for the OBDH and DHS modems two possible synchronisation patterns are considered: the previous one and LOLOLOHOHOHO.

This unit performs the following tasks:

- Detects Low logic levels in the nominal and redundant input signals.
- Extracts the recovered clock from the nominal and redundant buses.

This unit performs a first demodulation of the incoming signals as a series of '0's and '1's. It must be pointed out that whenever NIDS1n or RIDS1n is '0', it considers there is a '1' in the corresponding bus, and if NIDS2n or RIDS2n is '0', it considers there is a '0'. Both buses are sampled by using the LOSC input signal as a clock. It must be an 8 MHz signal in order to obtain a correct decoding. It must be noted that the four signals are captured two times in order to avoid metastability. This series of data will be further decoded in order to obtain the synchronisation sequence and the valid or invalid '0's and '1's transmitted through the bus.

The nominal and redundant clocks are generated by simply performing the logical and of the NIDS1n and NIDS2n signals for the nominal clock, and the RIDS1n and RIDS2n signals for the redundant clock. The behaviour of this unit is illustrated in the figure below.

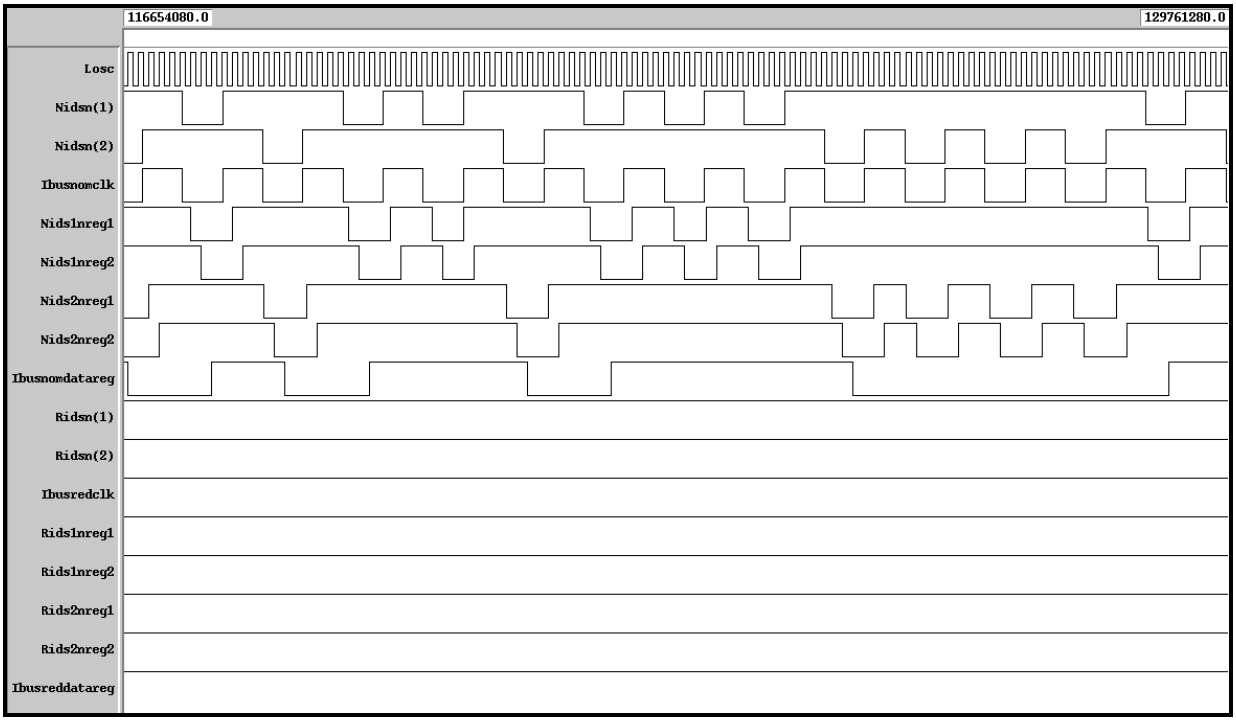


Figure 6.5 Clock and Data recovering on the I-bus.

6.2.2 Watchdog and Bus Selection

CLOCK Signals
- LOSC

Input Signals
- OBDHpDHSn
- CASCADING
- SIMUL
- CITSEL
- CTpRTn
- SELECTINN
- SELECTINR
- IBusNomCLK
- IBusRedCLK
- IBusNomDataReg
- IbusRedDataReg

Output Signals
- DetectINomSync
- DetectIRedSync
- IBusNomSync
- IBusRedSync
- INomActive
- IRedActive
- ACTIVEN
- ACTIVER
- SelNomIBus
- SELECTOUTN
- SELECTOUTR
- TIMEOUTn

This unit performs the following tasks:
- Decodes the raw data bits coming from the Clock and Data recovering unit as series of bits synchronous with the Recovered clocks.
- Detects Synchronisation patterns.
- Activates the INomActive or IRedActive signals by detecting three consecutive sync patterns.
- Deactivates the INomActive and IRedActive signals by loosing three consecutive sync patterns. (Watchdog function).
- Generates the ACTIVEN and ACTIVER BSS output signals.

- Activates the Selection flag SelNomIBus (SelNomIBus = '1', nominal set of buses selected; SelNomIBus = '0', redundant buses selected) according to the value of the INomActive and IRedActive signals in RT mode or of the Selection inputs in CT mode (see also SIMUL and CASCADING discussion below).
- Generates the SELECTOUTN and SELECTOUTR BSS output signals.
- Activates the TIMEOUTn signal when transition occurs in bus selection.

This unit is composed of two submodules that perform some of the previous tasks for the nominal and redundant buses. Bellow appears the description of one of the submodules. The other has the same behaviour but applied to the redundant bus.

The IBusNomCLK nominal I-bus incoming clock signal is doubly registered in this unit in order to avoid metastability. After that, rising edges on this signal are checked. When detected, the series of incoming data bits IbusNomDataReg are stored in a shift register (6 bits in depth), called IbusNomSync. This shift register is checked in order to detect synchronisation patterns. When detected, a pulse is generated in the DetectINomSync signal. It must be noted that for OBDH and DHS the "111000" and "000111" patterns are valid. When three consecutive sync patterns are detected with the correct separation (64 IBusNomCLK clock pulses), the INomActive flag is activated. A watchdog counter is reset every time a Sync pattern is detected. Otherwise, the INomActive signal is deactivated.

Bus Activity detection and Bus selection behaviour is shown in the following figure.

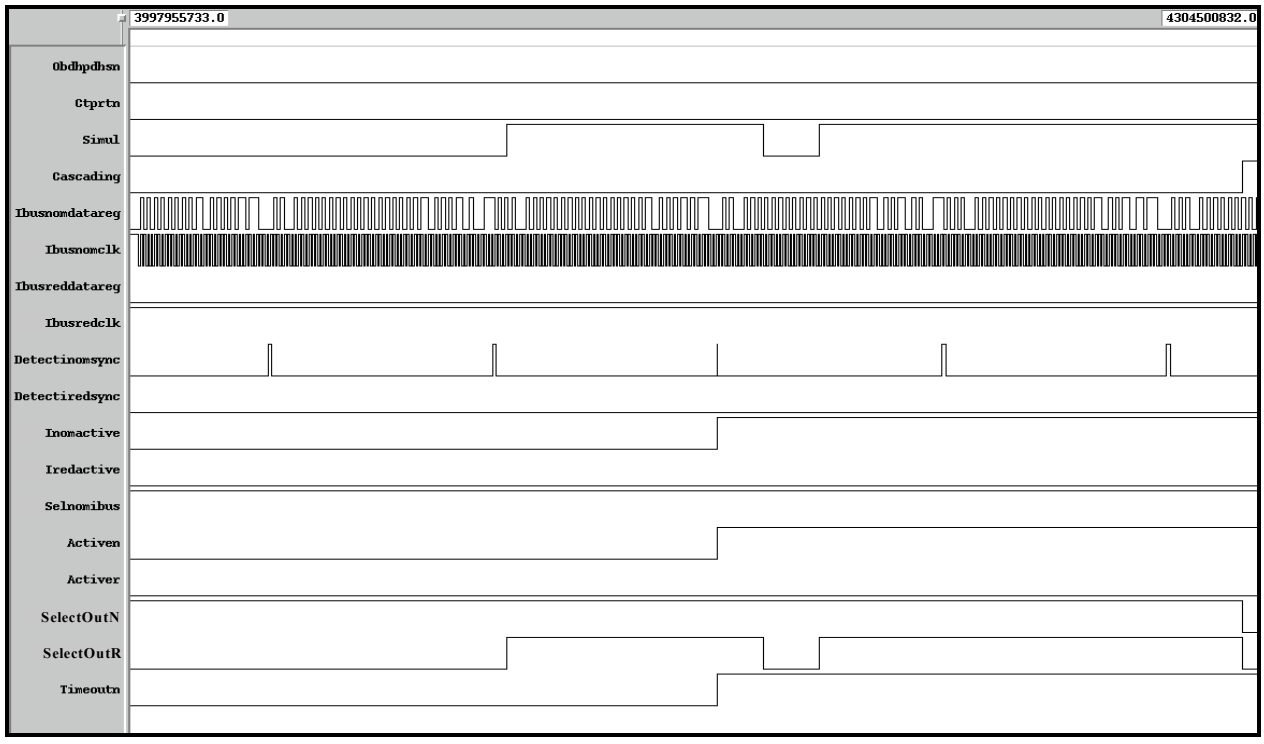


Figure 6.6 Bus activation and selection.

The selection of the buses depends on the function mode (OBDH or DHS and CT or RT), on the activity detected and also on the inputs CITSEL, SELECTINN, SELECTINR, CASCADING and SIMUL. In the following tables can be seen how the selection is made:

CASCADING Signal	Bus selection status
L	Normal selection matrix
H	Both buses deselected

Table 6.1 Bus selection in CASCADING configuration.

If CASCADING is not asserted the bus selection is performed by taken into account the SIMUL input in the following way:

SIMUL Signal	Bus selection status
L	Normal selection matrix
H	Both buses selected

Table 6.2 Bus selection according to SIMUL.

If CASCADING and SIMUL are not asserted bus selection is performed as shown in tables 6.3, 6.4 and 6.5.

When basic modem operates in RT mode, selection of the buses is performed according to the next table:

Nominal Bus	Redundant Bus	Selected Bus
Inactive	Inactive	N-Bus
Active	Inactive	N-Bus
Inactive	Active	R-Bus
Active	Active	N-Bus

Table 6.3 Selection matrix in RT mode

When DHS basic modem operates in CT mode, selection of the buses is performed according to the following table:

Nominal Bus Selection Request Input	Redundant Bus Selection Request Input	Selected Bus
Not-asserted	Not-asserted	N-Bus
Asserted	Not-asserted	N-Bus
Not-Asserted	Asserted	R-Bus
Asserted	Asserted	N-Bus

Table 6.4 BSS selection matrix in CT modem

When OBDH basic modem operates in CT mode, selection of the Buses is performed according to the table 6.5.

CITSEL Signal	Selected Bus
L	R-Bus
H	N-Bus

Table 6.5 Selection matrix in OBDH CT modem.

6.3 OBDH BUS TERMINAL (DBI)

All these interfaces are available only for OBDH modem, if the line OBDHpDHSn is at low logic level all the outputs remain at low logic level.

6.3.1 BAR1 Port (RIR)

CLOCK Signals
- IbusCLK

Input Signals
- OBDHpDHSn
- CTpRTn
- TIMEOUTn
- INomActive
- IRedActive
- SelNomIBus
- CountRIRBits
- DetectISync
- IBusSync
- IBusDataReg

Output Signals
- RIRCLK
- RIRDATA
- RIRVAL
- RIRSYNC

This interface behaves according to AD_1 (page 7). But RIRVAL and RIRSYNC signals are delayed ¼ bit. The reason is because this interface is synchronous with the IBusCLK signal. Thus RIRVAL changes in the Rising edge of clock (not in the falling edge). Otherwise, an asynchronous design must be performed.

The second difference with the AD_1 applicable document is that the RIR signals are going to be divided in two sets of input and output signals. It must be noted that they were bidirectional signals in the OBT chip. The reason for this change against the original specification is that technological problems may occur due to the bidirectional clock signal RIRCLK.

This unit performs the following tasks, when active. It detects DetectISync, which indicates that a synchronisation pattern has been received. If so, a counter is started (CountRIRBits) and the signal RIRCLK is put in phase according to page 7 in AD_1. After that, a sequencer is activated. Data received through the IbusSync shift register are checked in order to:

- Detect Synchronisation patterns (Note that DetectISync signal occurs after RIRSYNC).
- Decodes incoming data and put it in the RIRDATA output signal.

- Detects invalid bits and deasserts, if necessary, the RIRVAL output signal.
-

This interface is active only under the following conditions:

- CTpRTn input signal is Low.
- OBDHpDHSn input signal is High.
- There is activity in one of the I-buses (nominal or redundant).
- The TIMEOUTn signal is asserted at high level logic state.

When one of the previous conditions doesn't occur the RIR output signals remains at inactive low level logic state.

The behaviour of this unit is reflected in the following figure.

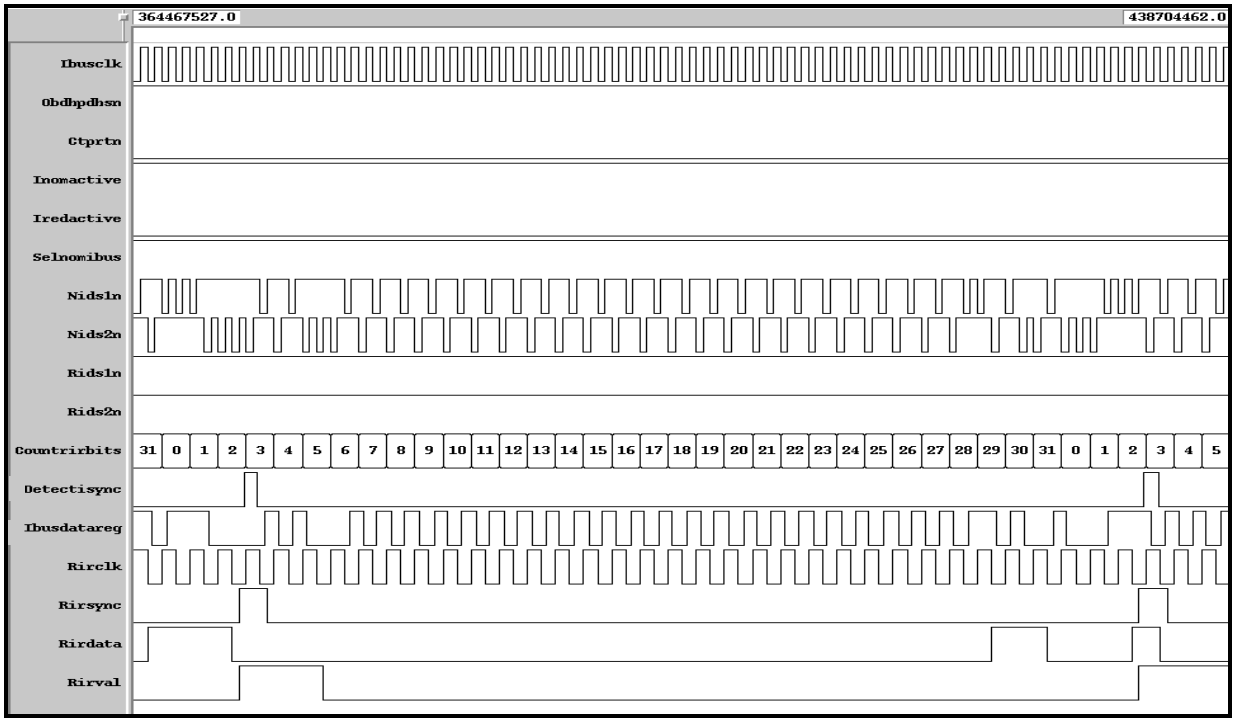


Figure 6.7 BAR1 Timing

6.3.2 BAR2 Port (RRR/CRR)

CLOCK Signals
- LOSC
- RbusCLK (internal)

Input Signals
- OBDHpDHSn
- NRDS1n
- NRDS2n
- RRDS1n
- RRDS2n
- SelNomIbus
- CRRINIT

Output Signal
- RBusDataReg
- RBusData
- CRRCLK
- CRRDATA
- CRRVAL

It must be noted that CRRVAL signal is delayed by ¼ bit against the specification in AD_1, page 8. The reason for this change is to perform a synchronous design that otherwise would be impossible.

This unit receives data from the R-bus and puts the incoming data in the RRR/CRR output signals.

This unit is divided in two submodules: the first one recovers data and clock from the R-bus Litton code, the second submodule generates the RRR/CRR signals. This interfaces works only when the signal OBDHpDHSn is at high level.

Clock and Data Recovering submodule

R-bus clock is recovered by performing the logical “and” of the XRDS1n and XRDS2n input signals. The SelNomIBus input signal selects the nominal or redundant bus. R-bus Data is recovered by sampling the XRDS1n and XRDS2n input signals synchronously with the LOSC input oscillator. These signals are doubly registered in order to avoid metastability. The incoming data is put in the RBusDataReg register. This signal is captured in the RbusData register clocked by the RBusCLK internal clock. Thus, decoded data is produced synchronously with the RBusCLK signal.

RRR/CRR interface submodule

In this submodule the CRRINIT input signals asynchronously resets the CRRCLK, CRRDATA, and CRRVALID signals. Then this submodule is ready to start a transmission.

CRRCLK and CRRDATA are produced in the falling edge of RBusCLK. CRRVAL is produced in the Rising edge of the RBusCLK clock. Thus, CRRVAL signal is delayed by ¼ bit against the specification in AD_1, page 8. CRRDATA is generated by sampling the RbusData signal value. CRRVAL is generated by checking the RbusData and RbusDataReg registers, in order to detect invalid Litton codes.
The behaviour of this unit is illustrated in the figure below.

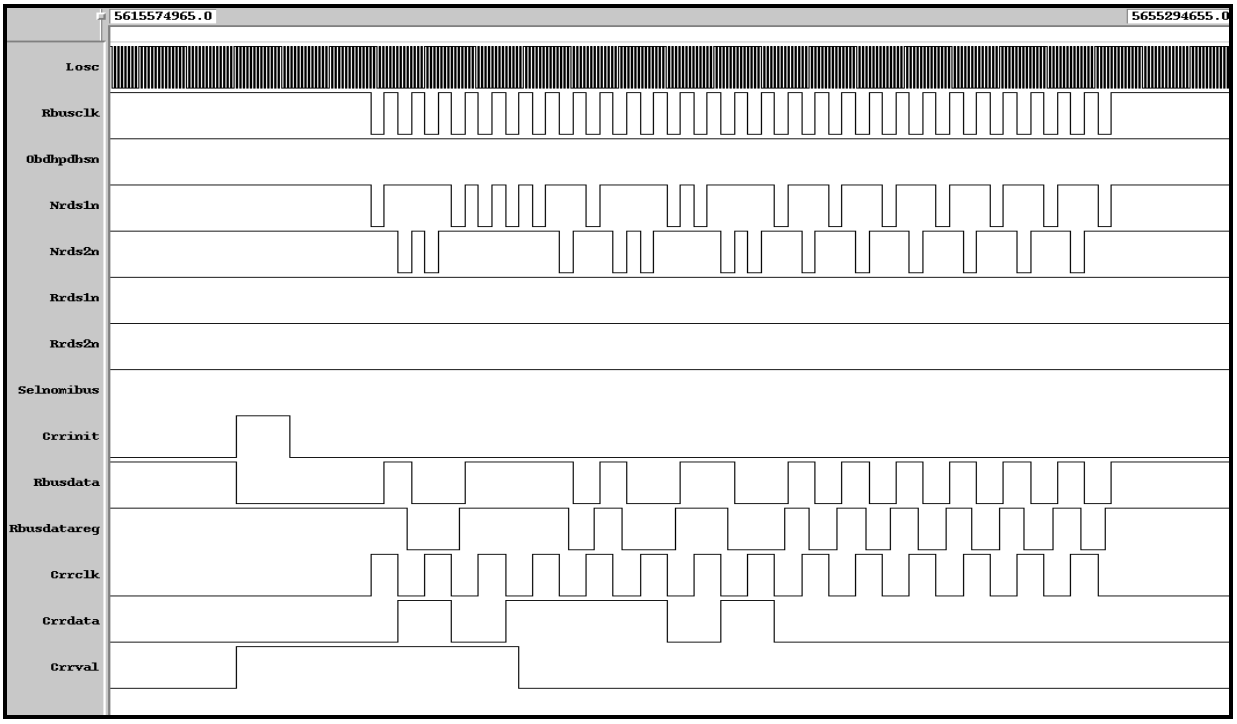


Figure 6.8 BAR2 Timing.

6.3.3 BAR3 Port (RBR/CBR)

CLOCK Signals
- LOSC
- BTBusCLK (internal)

Input Signals
- OBDHpDHSn
- CTpRTn
- NBDS1n
- NBDS2n
- RBDS1n
- RBDS2n
- SelNomIbus
- CBRINIT

Output Signals
- BTBusDataReg
- BTBusData
- CBRCLK
- CBRDATA
- CBRVAL

This module makes its function when the modem works as OBDH.

It must be noted that CBRVAL signal is delayed by ¼ bit against the specification in AD_1, page 8. The reason for this change is to perform a synchronous design that otherwise would be impossible.

This unit receives data from the BT-bus and puts the incoming data in the RBR/CBR output signals.It is divided in two submodules: the first one recovers data and clock from the BT-bus Litton code, the second submodule generates the RBR/CBR signals.

Clock and Data Recovering submodule

R-bus clock is recovered by performing the logical “and” of the XBDS1n and XBDS2n input signals. The SelNomIBus input signal selects the nominal or redundant bus. Then, the BTBusCLK internal clock signal is generated.

BT-bus Data is recovered by sampling the XBDS1n and XBDS2n input signals synchronously with the LOSC input oscillator. These signals are doubly registered in order to avoid metastability. SelNomIBus input signals selects the nominal or redundant input data. Thus the incoming data is put in the BTBusDataReg register. This signal is captured in the BTbusData register clocked by the BTBusCLK internal clock. Then, decoded data is produced synchronously with the BTBusCLK signal.

RBR/CBR interface submodule

In this submodule the CBRINIT input signals asynchronously resets the CBRCLK, CBRDATA, and CBRVALID signals and force a known state before the transmission is performed.

CBRCLK and CBRDATA are generated in the falling edge of BTBusCLK. CBRVAL is produced in the Rising edge of this clock. Thus, CBRVAL signal is delayed by ¼ bit against the specification in AD_1, page 8. CBRDATA is generated by sampling the BTbusData signal value. CBRVAL is deasserted by checking the BTbusData and BTbusDataReg registers in order to detect invalid Litton codes.
The behaviour of this unit is illustrated in the figure below:

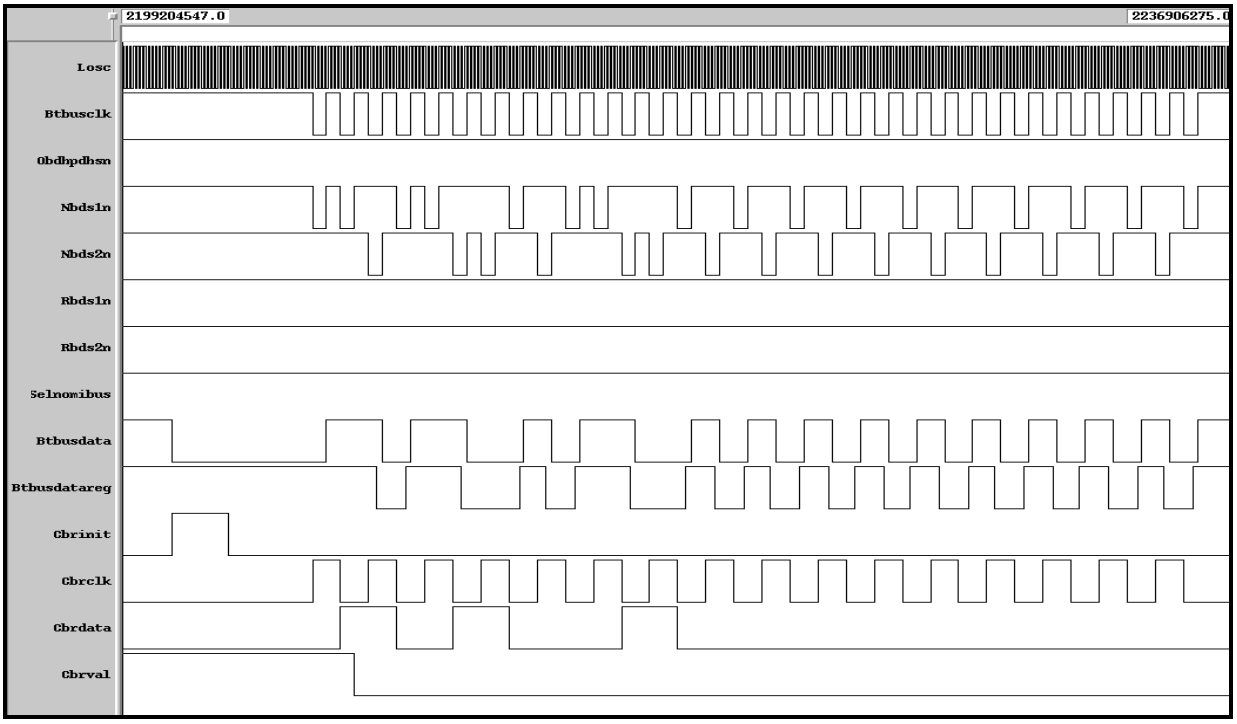


Figure 6.9 BAR3 Timing.

6.3.4 BAT1 Port (RRT/CIT)

CLOCK Signals
- CITMOD
- IBusCLK.

Input Signals
- OBDHpDHSn
- CTpRTn
- CASCADING
- SIMUL
- SelNomIbus
- RRTEN
- RRTDATA
- CITCLK
- CITDATA
- CITSYNC
- CITSEL

Output Signals
- NRE
- RRE
- RR1n
- RR2
- RR3n
- RR4

This unit implements exactly the functionality described in [AD_1] document (pages 5 and 6). The only difference is that the signal CITINV and its function, explained in that document, has been remove from the design described in this pages.

This unit is composed of three submodules:
- RRT interface module.
- CIT interface module.
- I/R-Bus transmission interface module.

RRT interface module

This submodule is active when the ASIC performs the function of an OBDH's remote terminal (OBDHpDHSn = '1' and CTpRTn = '0'). The RRTEN and RRTDATA incoming signals are sampled with the IBusCLK clock. When RRTEN is at high level, the data presented in RRTDATA is passed to the I/R-Bus transmission interface submodule.

CIT interface module.

This submodule is on for an OBDH bus controller (OBDHpDHSn = '1' and CTpRTn = '1') and the CITMOD input signal is the active clock. The CITCLK signal has two times the period of CITMOD and is used to enable capture of CITDATA when stable. When a pulse is introduced by the CITSYNC input a synchronisation pattern is sent to the bus. If the CITDATA value in that moment is '0' then is sent the normal sync pattern ("HOHOHOLOLOLO"), otherwise if it's '1' is generated the inverted pattern ("LOLOLOHOHOHO"). The signal CITSEL is used to select the bus to transmit to in CT mode.

I/R-Bus transmission interface module

This unit receives series of data from the RRT and CIT interfaces and puts them in the RR1n, RR2, RR3n, and RR4 output signals.

It controls also the generation of the NRE and RRE output signals. NRE and RRE take the following values: When CASCADING = '1' then both signals are always deasserted.

In CT mode (CTpRTn = '1'):
NRE is '1' if SIMUL is high, otherwise it takes the SelNomIbus value.
RRE is '1' if SIMUL is high, otherwise it takes the inverted value of SelNomIbus.
In RT mode (CTpRTn = '0'):
NRE and RRE take the same value as in CTpRTn but they are active high just when RRTEN is high.

I/R bus transmitted data is selected considering the following possibilities: OBDHpDHSn is '1' or '0' and CTpRTn is '1' or 0. In the following table selected data is described.

OBDHpDHSn	CTpRTn	Source
1	1	CITDATA
1	0	RRTDATA
0	1	ITSDATA
0	0	RTSDATA

Table 6.6 I-Bus source Data.

It must be pointed out that incoming data will be formatted in order to send the corresponding valid Litton codes.

The behaviour of RRT and CIT interfaces is depicted in the two figures below.

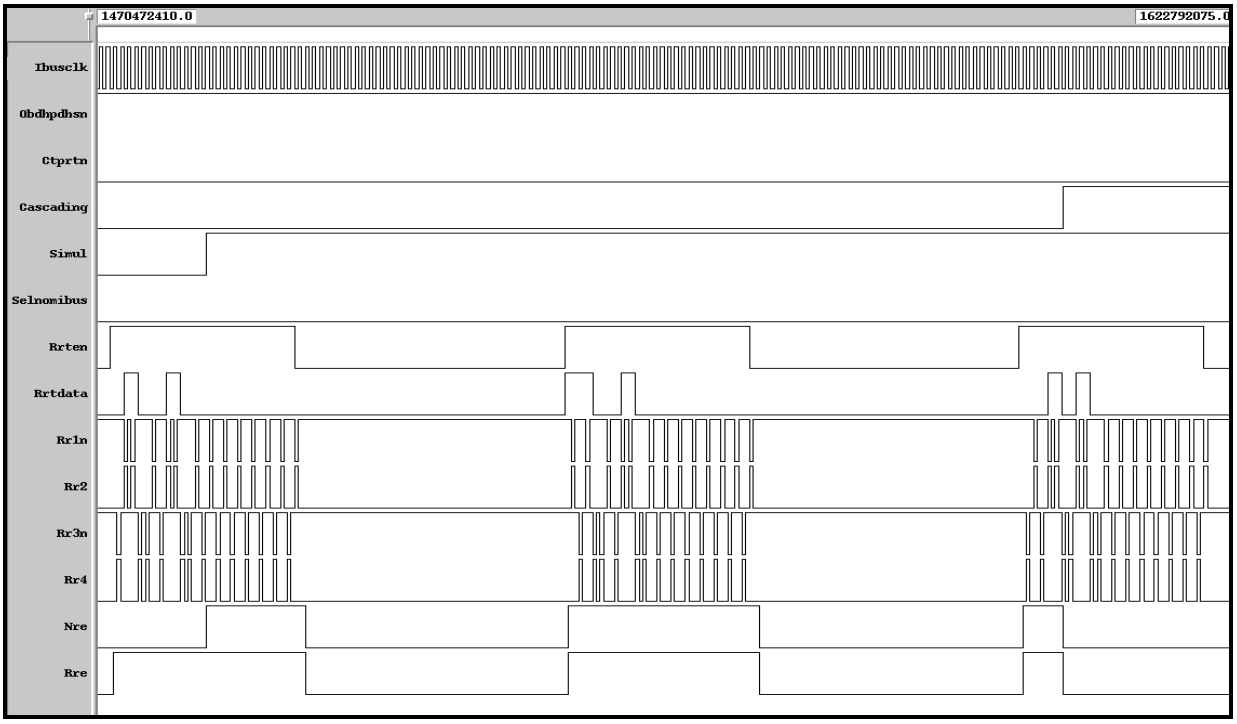


Figure 6.10 RRT Interface Timing.

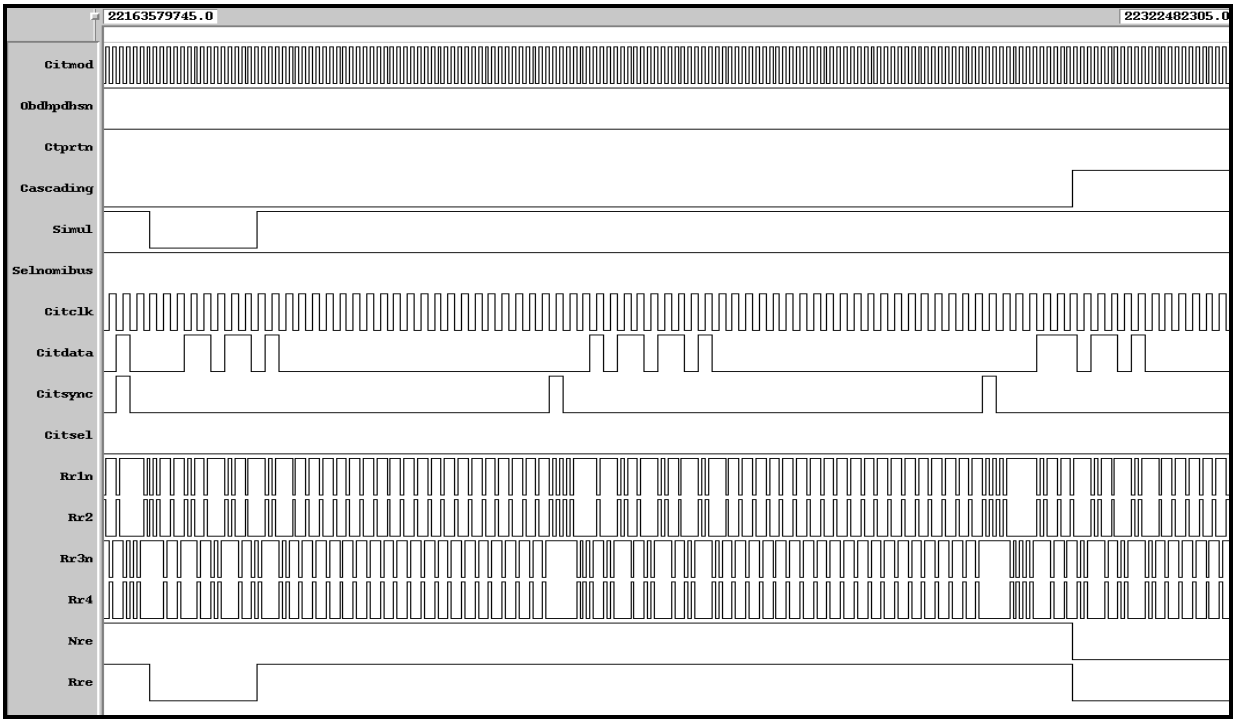


Figure 6.11 CIT Interface Timing.

6.3.5 BAT2 Port (RBT/CBT)

CLOCK Signals
- CITMOD
- IBusCLK.

Input Signals
- OBDHpDHSn
- CTpRTn
- CASCADING
- SIMUL
- SelNomIbus
- RBTEN
- RBTDATA

Output Signals
- NBE
- RBE
- RB1n
- RB2
- RB3n
- RB4

This unit implements exactly the functionality described in [AD_1] document (page 6).

This unit is composed of three submodules:
- RBT interface module.
- CBT interface module.
- BT-Bus transmission interface module.

It must be noted that although the RBTEN and CBTEN, and RBTDATA and CBTDATA input signals are exactly the same, there are two different interfaces because RBT works with the IBusCLK clock, while CBT works with the CITMOD clock. Functionally they are identical.

RBT interface module

This submodule is active when the ASIC works as an OBDH bus remote terminal (OBDHpDHSn = ‘1’ and CTpRTn = ‘0’). The RBTEN and RBTDATA incoming signals are sampled with the IBusCLK clock. If RBTEN is ‘1’ then the data sampled in RBTDATA is passed to the BT-Bus transmission interface submodule.

CBT interface module

When the control signals OBDHpDHSn and CTpRTn take the high logic value, the signals CBTEN and CBTDATA are sampled with the CITMOD clock. Sampled data presents in CBTDATA is passed to the BT-Bus transmission interface submodule if CBTEN is high.

BT-Bus transmission interface module

This unit receives serial data from the RBT and CBT interfaces and puts them in the RB1n, RB2, RB3n, and RB4 output signals.

It controls also the generation of the NRE and RRE output signals. NRE and RRE take the following values:

If CASCADING = '1' then:
both signals are '0'.
Otherwise, if SIMUL = '1' then:
NBE and RBE are '1' if RBTENABLE is high, in RT mode, or CBTENABLE, in CT mode.
If CASCADING and SIMUL are at low level:
NBE takes SelNomIbus value, if RBTENABLE is high in RT mode, or CBTENABLE in CT mode.
RBE takes the inverted SelNomIbus value, if RBTENABLE is high in RT mode, or CBTENABLE in CT mode.

BT_Bus transmitted data is selected considering the following possibilities: OBDHpDHSn is '1' or '0' and CTpRTn is '1' or '0'. In the following table selected data is described.

OBDHpDHSn	CTpRTn	Source
1	1	CBTDATA
1	0	RBTDATA
0	1	BTTSDATA
0	0	BTTSDATA

Table 6.7 BT-Bus source Data.

It must be pointed out that incoming data will be formatted in order to send the corresponding Litton codes.

The behaviour of RBT and CBT interfaces is depicted in the figure below.

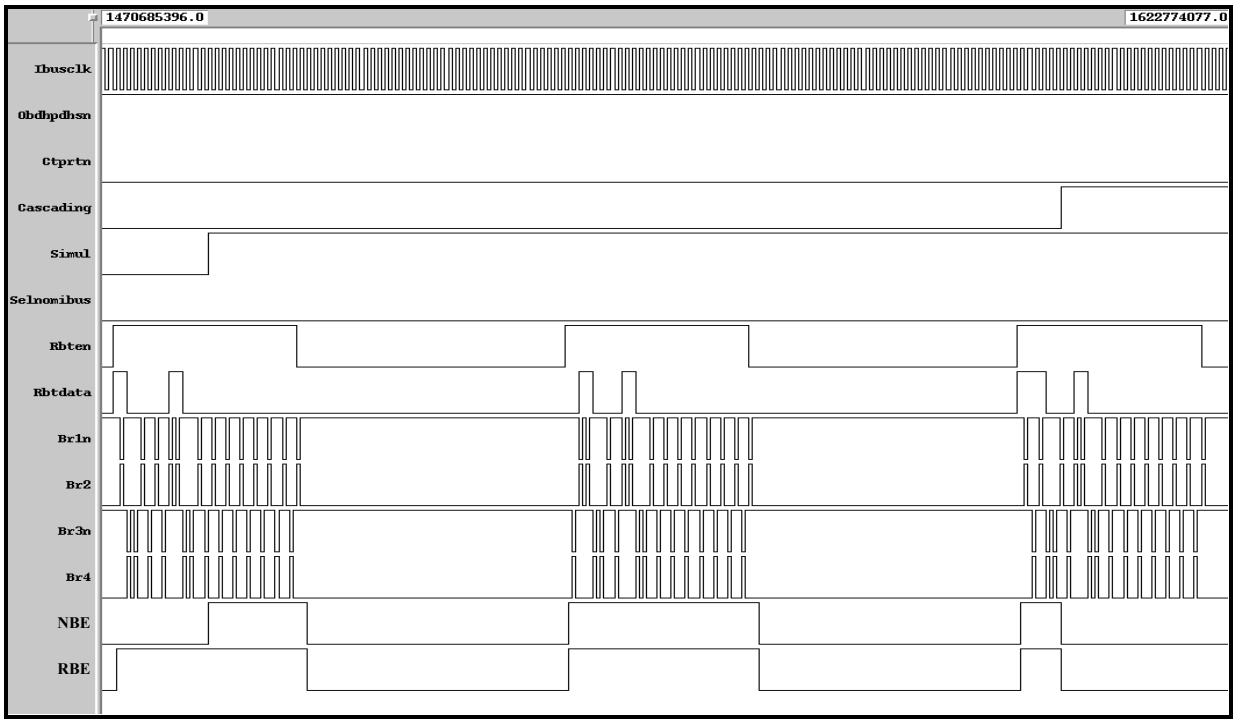


Figure 6.12 RBT Interface Timing.

6.3.6 IUB

CLOCK Signals

- IBusCLK.

Input Signals

- OBDHpDHSn
- CTpRTn
- TIMEOUTn
- RIRDATA
- RIRVAL
- RIRSYNC
- TAV
- OOINH
- EXTMLA(1:2)
- EXTFMT
- TA(0:5)
- ANSIN
- DIGIN

Output Signals

- CTCLK
- IRCLK
- TRCLK
- BCP
- BCPVAL
- MLADD
- MLDATA
- CHADD
- MOAND
- MOANS
- MODBL
- MODS8
- MODS16
- MOHL
- MOLC
- MOSC
- MOBT
- PC
- ANCLK
- SOC
- SH
- DATARRT
- ENRRT

This unit implements exactly the functionality described in [AD_1] document (pages 13 to 20).

The clock of this unit is the IBusCLK signal. However most of its functionality (with the exception of Analog acquisitions) is enabled by the RIRCLK high level value. Thus, most of the inputs and outputs are quasi synchronous with the RIRCLK falling edge. RIRCLK has not been used as a clock in order to minimise the clock signals. Also, an important reason is the analog acquisition timing, which requires a faster clock (such as IBusCLK).

This unit is divided in three main parts corresponding to the three slots, which configures a command. Also the actions taken depends on if RIRSYNC is active or not. In the following paragraphs the IUB sub-units behaviour is described in detail. It must be noted that this unit architecture is a three stages pipeline.

There is also an additional module composed by a shift register which loads the incoming RIRDATA, and a counter that counts if the number of bits transmitted per slot is correct.

If the input signal TAV is '0' then the RTU kernel is disabled, otherwise it performs its normal function.

RIRSYNC = '1'

It must be pointed out that whenever a RIRSYNC pulse is detected the CTCLK, IRCLK and TRCLK output clocks are put in their correct phase.

First Slot:

The Data accumulated in the shift register is decoded and stored in the corresponding control registers: BCP, DAddress, CHADD, ModeCode, MLADD, and TAddress. This unit checks the following information:

- The number of bit transmitted since the last RIRSYNC detection must be correct (i.e. 32).
- The parity must be correct.

When both conditions are true the BCP output signals take the value from the incoming data slot. BCPVal(3 to 5 or 3 to 6, depending on the EXTFMT value). BCPVAL takes the RIRVAL value. Otherwise it becomes zero. Thus, whatever error transmission resets the BCPVAL signal.

After that, it's checked that the incoming TA address matches the TA input signals. If so, and the previous slot was not a ML Command or a 16 bit digital serial acquisition (because they take two slots to complete) the control registers are loaded with the incoming data.

It must be noted that TA length varies according to the EXTMLA(1:2) inputs in the following way:

EXTMLA1	EXTMLA2	Behaviour
'0'	'0'	Normal 3 bits Memory load address field
'1'	'0'	Memory load address field is extended to 4 bits and bit 11 of the Interrogation will be treated as MLA(0).
-	'1'	Memory load address field is extended to 5 bits and bits 11 and 12 of the Interrogation will be treated as MLA(0) and MLA(1) respectively.

Table 6.8 MLA field composition.

Second Slot:

In the second Slot the control registers MLADD, Daddress, and ModeCode are passed to the second stage control registers.

Third Slot:

In this stage is stored the data to be send through the DATARRT line. Generally a 13 bits response is sent, but if the command is a 16 bit digital acquisition (MOS16) a 21 bits response is sent.

RIRSYNC = '0'

For all the stages, the CTCLK, IRCLK, and TRCLK signals are produced with the correct timing.

First Slot:

If the command is not a ML command the CHADD line is activated in the first CTCLK falling edge. The Modecode control register is decoded and the corresponding command signal is asserted high in the first CTCLK rising edge. If the command is MOLC the OOINH enable input signal is checked. If it's high no MOLC command will be generated. If the command is a digital acquisition DIGIN input data is stored.

If the command is a ML command, the MLADD and MLDATA output signals take the value stored. The 8 MSB data bits are serially shift through MLDATA.

Second Slot:

For ML commands the 8 LSB data bits are serially sent through the MLDATA line.

For 16 bits digital acquisitions (MOS16) the 8 LSB data bits are stored.

For analog acquisitions (MOANS and MOAND) the sequence of the ANCLK, PC, SH, SOC signals is generated. Also the ANSIN data line is internally stored and shifted.

Otherwise no action takes place.

Third Slot:

The ENRRT signal is asserted high if a correct command has been detected in the slot before the last one. Data to be sent or a '0' response is serially sent through the DATARRT line. If the command is a 16 bit digital acquisition a 21 bits response occurs. Otherwise, a 13 bits response takes place.

The timings for all the RTU kernel commands are displayed in the following figures. It has to be pointed out that some of them have similar waveforms. For example, MODS8 and MODBL can both be represented by the figure 6.15 (replacing the signal MODS8 by MODBL); MOSC, MOLC, MOHL and MOBT have the same timings as the one shown in figure 6.17.

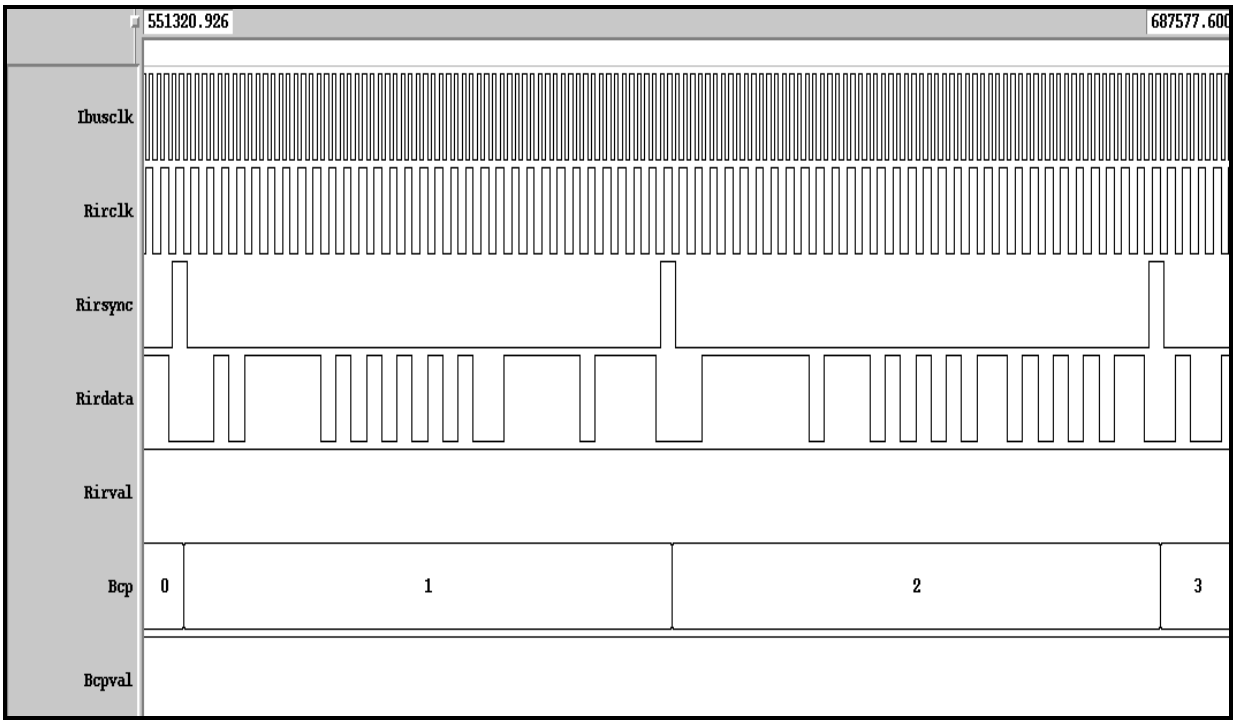


Figure 6.13 BCP Timing.

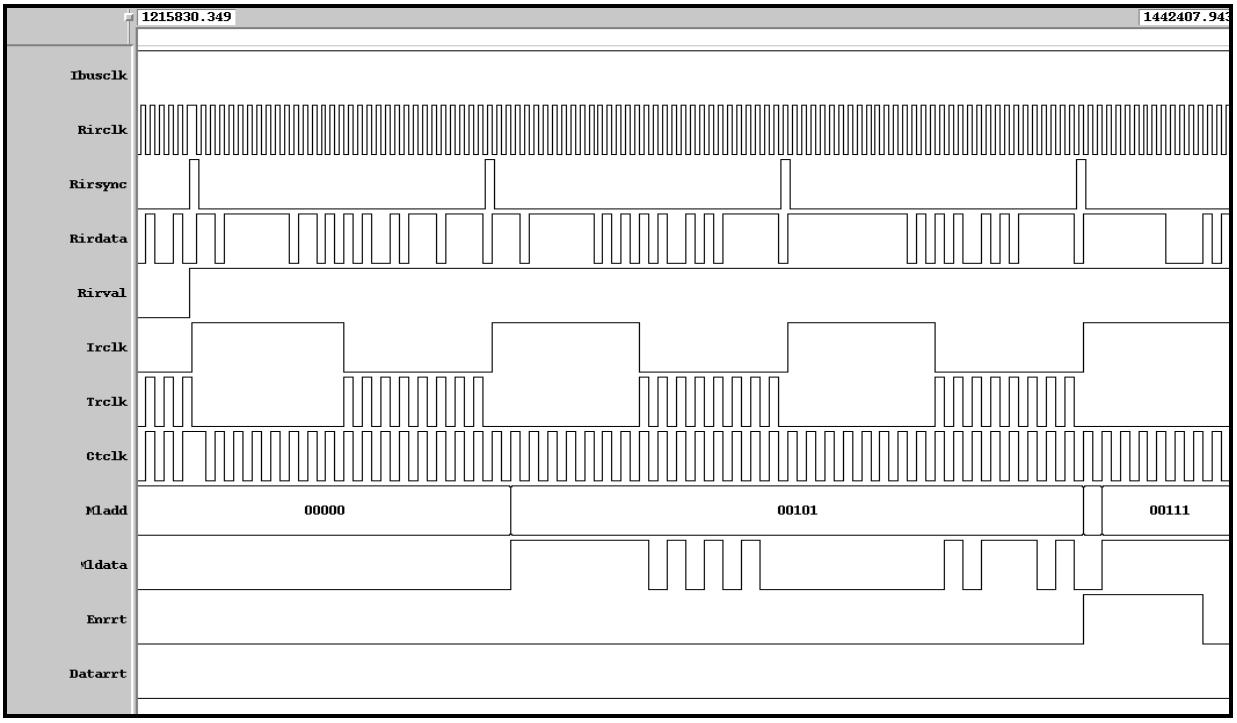


Figure 6.14 ML Command Timing.

Figure 6.15 MODS8 Timing.

Figure 6.16 MODS16 Timing.

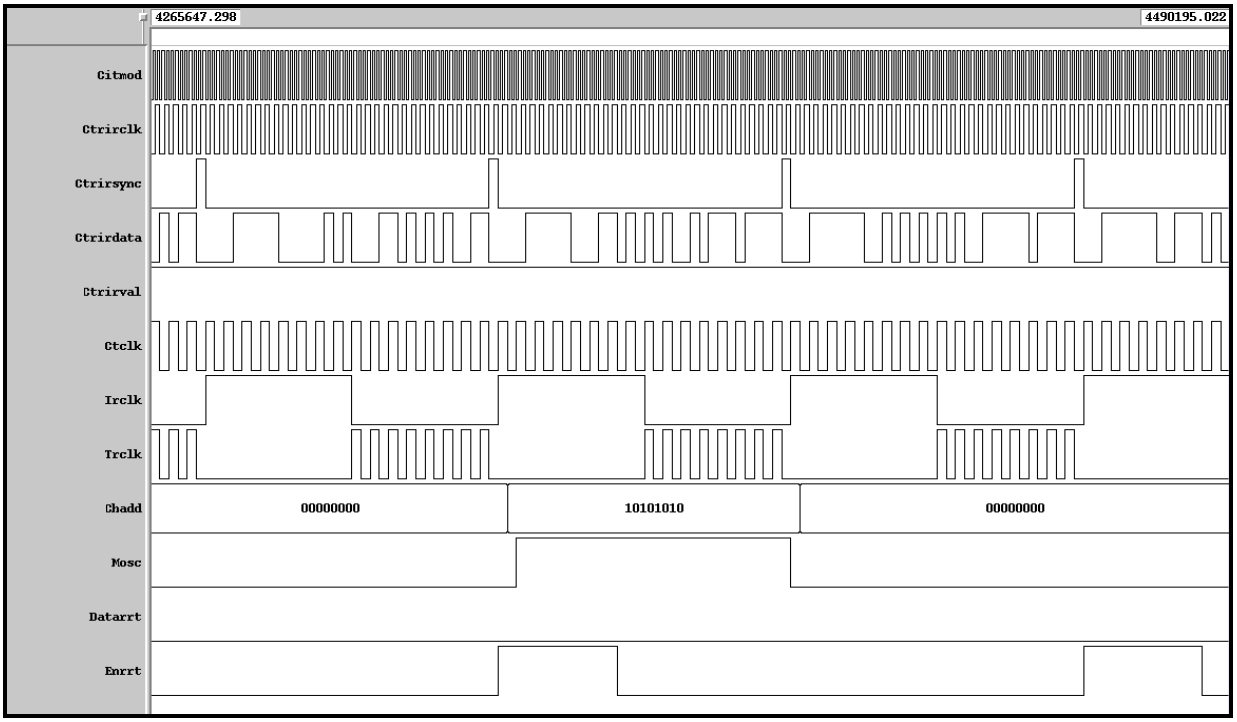


Figure 6.17 MOSC Timing.

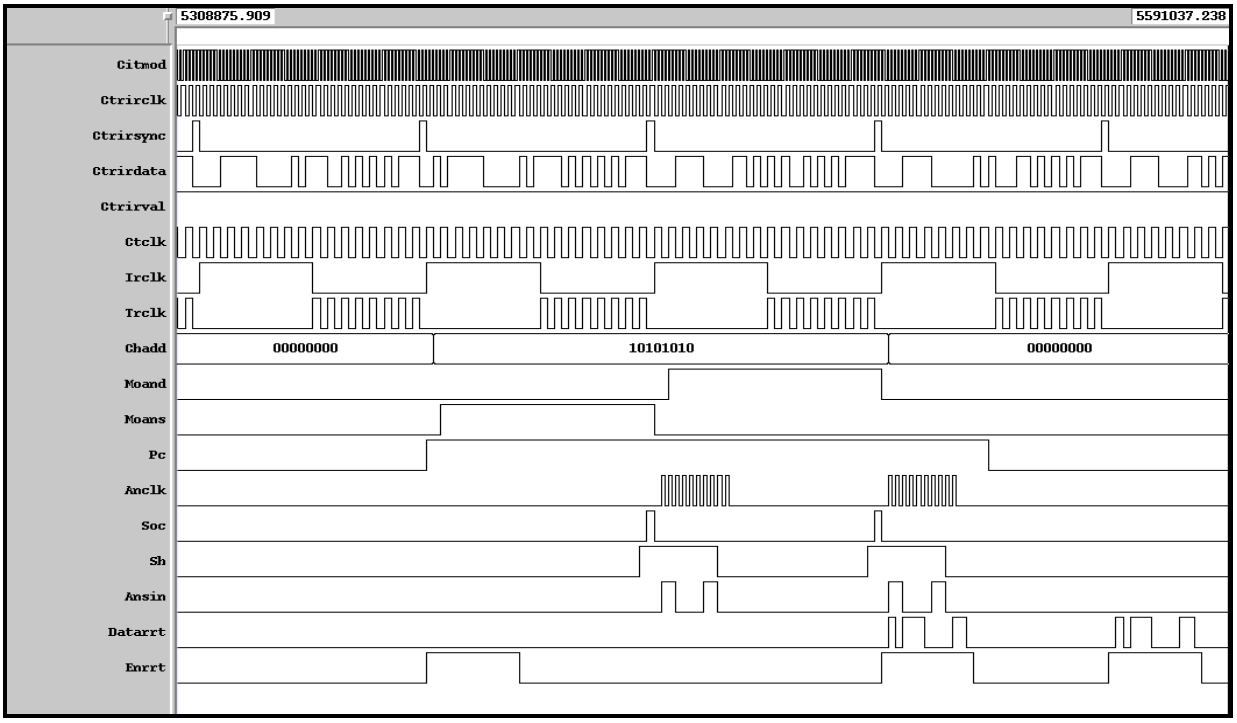


Figure 6.18 MOAND and MOANS Timing.

6.4 DHS BUS TERMINAL

6.4.1 BTS Service

CLOCK Signals

- MASTEROSCIN
- IBusCLK.

Input Signals

- OBDHpDHSn
- CTpRTn
- DetectISync
- BTSENABLEIN

Output Signals

- BTSCLKOUT
- BTSSYNC

This unit contains two submodules that implement the equivalent functionality for CT and RT. The CT submodule operates with the input clock signal MASTEROSCIN. The RT submodule operates with the IBusCLK recovered clock. After that CTpRTn selects the source for the BTSCLK_OUT and BTSSYNC output signals.

The CT submodule produces the BTSCLK_OUT output simply by dividing by 4 the MASTEROSCIN clock input. Every 32 bits period the BTSSYNC output signal is asserted high during a BTSCLK_OUT cycle. This submodule is only active if the BTSENABLEIN input signal is asserted high. The first BTSCLK_OUT pulse is generated between 3 and 4 MASTEROSCIN cycles after BTSENABLE becomes high level active.

The RT submodule produces the BTSCLK_OUT output simply by dividing by 2 the IBusCLK clock. BTSSYNC is asserted high during a BTSCLK_OUT cycle at bit 31 of the slot. The start of the slot is detected by means of the synchronisation pattern (indicated by the DetectISync signal).

The behaviour of this unit is illustrated in the following figure:

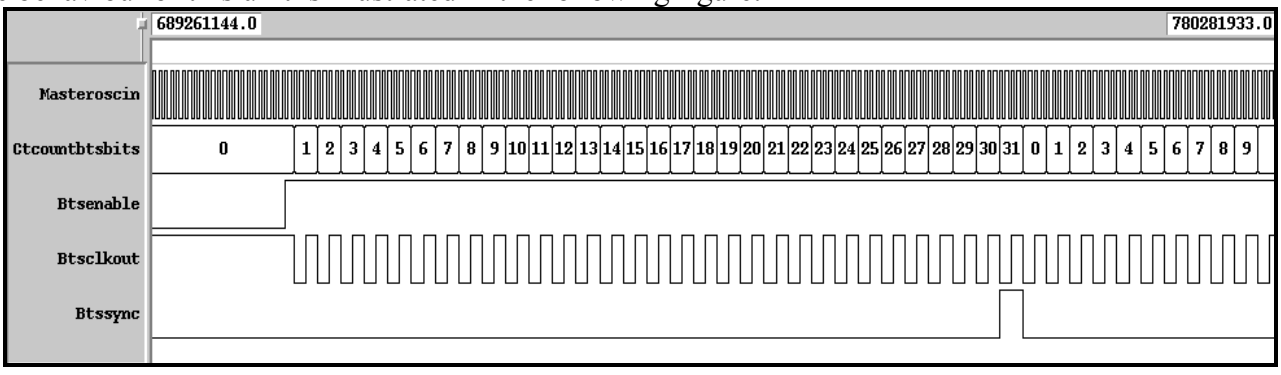


Figure 6.19 CT BTS Timing.

6.4.2 ITS Service

CLOCK Signals

- MASTEROSCIN.

Input Signals

- OBDHpDHSn
- CTpRTn
- BTSSYNC
- ITSENABLE
- ITSDATAIN

Output Signals

- ITSCLKOUT
- EnableITSTransmit
- ITSDATAOUT

This unit implement the behaviour described in AD_2 (page 36 paragraph 7.3.2.1). It starts its function when OBDHpDHSn = '0' and CTpRTn = '1'. ITS begins to transmit when there is a pulse in BTSSYNC and ITSENABLE is asserted. When the BTS service is not enabled, BTSSYNC is not generated and the ITS service doesn't work. Also if ITSENABLE is deasserted the transmission will finish in the next BTSSYNC pulse. Transmission will always be synchronized with slots boundaries. The incoming data ITSDATAIN is captured in ITSDATAOUT internal signal and sent to the I/R bus transmission module of the BAT1 unit. EnableITSTransmit signal enables transmission in the I/R-bus. The three first data captured in ITSDATAOUT after the BTSSYNC are the corresponding to the synchronization pattern. It will be normal or inverted depending on the data that appears in ITSDATAIN when the synchro pulse arrives ('1' for inverted pattern).

The behaviour of this unit is illustrated in the figure below.

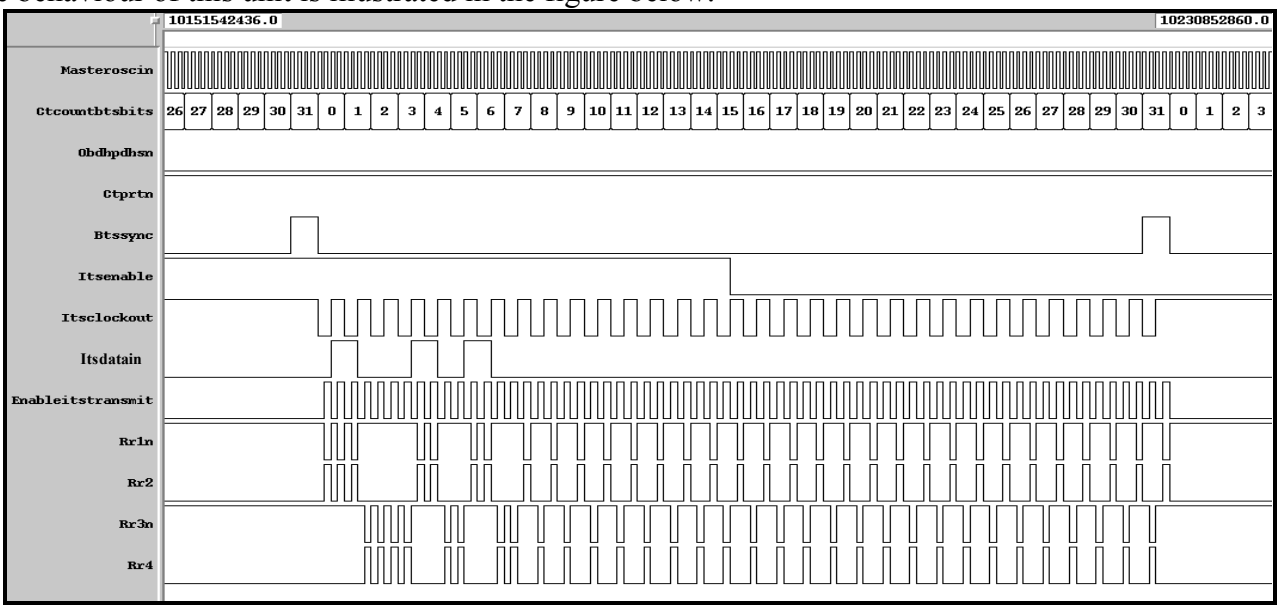


Figure 6.20 ITS Timing.

6.4.3 IRS Service

CLOCK Signals

- IBusCLK.

Input Signals

- OBDHpDHSn
- CTpRTn
- IRSENABLE
- IBusDataReg

Output Signals

- IRSDATAOUT
- IRSVALIDOUT
- IRSCLKOUT

This unit implements the behaviour described in AD_2 (page 39 paragraph 7.3.2.2). It is only active in RT mode. When IRSENABLE goes to high-level logic state (in the falling edge of the BTSSYNC signal) the reception process starts. Data Captured in the Clock and recovering unit (paragraph 7.2.1) is decoded and sent to the IRSDATAOUT output signal. IRSVALIDOUT output signal indicates whether a valid or invalid Litton code has been received. IRSCLOCK is generated by dividing by 2 the IBusCLK internal clock. IRSDATAOUT output and IRSVALID output signals are sent synchronously with the IRSCLOCK output signal (they are generated in rising edge).

The behaviour of this unit is illustrated in the figure below.

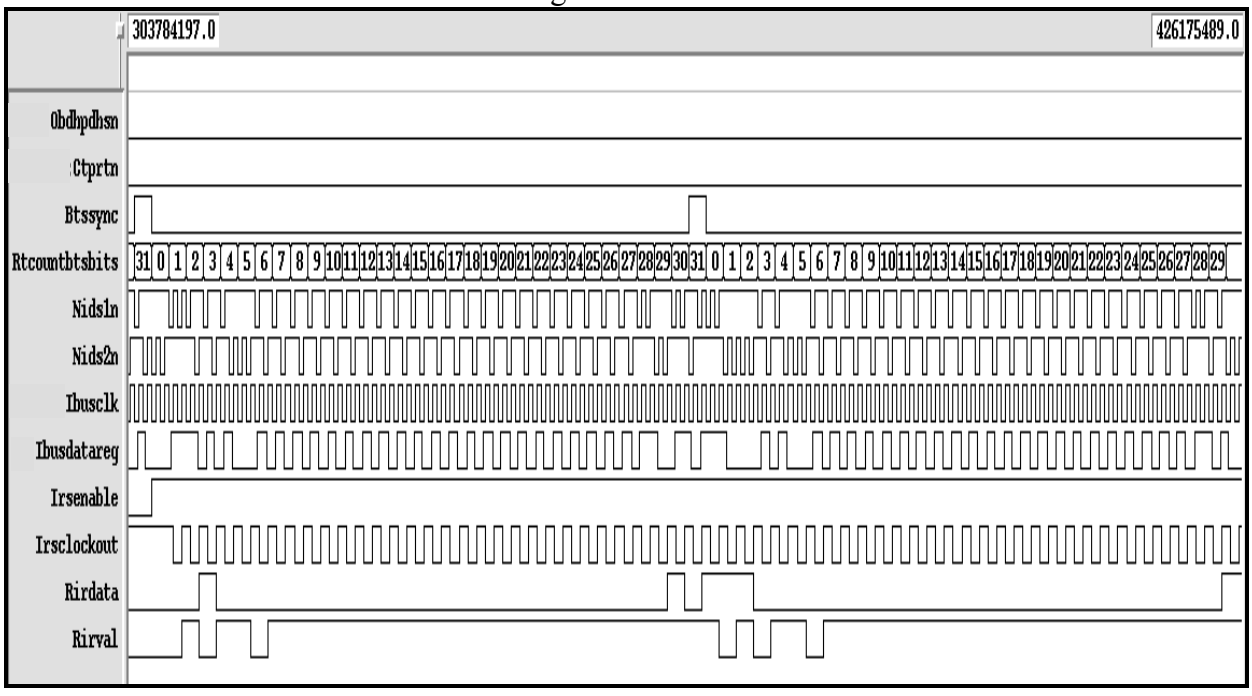


Figure 6.21 IRS Timing.

6.4.4 RTS Service

CLOCK Signals

- IBusCLK.

Input Signals

- OBDHpDHSn
- CTpRTn
- RTSENABLE
- RTSDATAIN

Output Signals

- RTSCLKOUT
- RTSDATAOUT
- RTSENABLEReg

This unit implement the behaviour described in AD_2 (page 41 paragraph 7.3.2.3), when the ASIC acts as a remote terminal of a DHS bus (OBDHpDHSn = '0' and CTpRTn = '0').

If RTSENABLE is at high-level logic state, the transmission process starts. The incoming data RTSDATAIN is sent to the I/R bus transmission module of the BAT1 unit, after being captured into the RTSDATAOUT internal signal. Then this data is multiplexed with other sources in the I/R bus generation module. RTSENABLEReg signal enables transmission in the I/R-bus.

The behaviour of this unit is illustrated in the figure below.

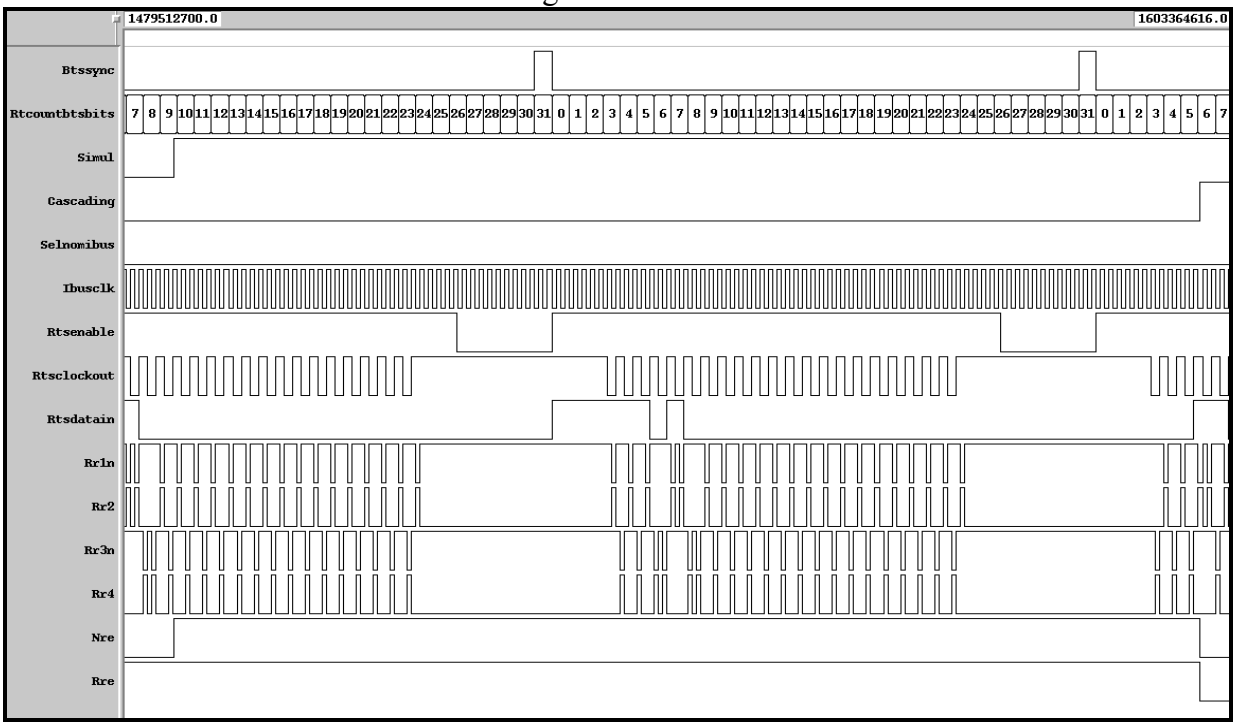


Figure 6.22 RTS Timing.

6.4.5 RRS Service

CLOCK Signals

- RBusCLK.

Input Signals

- OBDHpDHSn
- CTpRTn
- RBusDataReg
- RRSENABLE

Output Signals

- RRSDATAOUT
- RRSVALIDOUT
- RRSCLKOUT

This unit implements the behaviour described in AD_2 (page 43 paragraph 7.3.2.4). It must be noted that this unit works in DHS modem and in both CT and RT modes. If RRSENABLE is at high-level logic state, the reception process start. Data Captured in the Clock and recovering unit (paragraph 7.2.2) is decoded and sent to the RRSDATAOUT output signal. RRSVALIDOUT output signal indicates whether a valid or invalid Litton code has been received. RRSCLKOUT is generated by dividing by 2 the RBusCLK internal clock. RRSDATAOUT and RRSVALIDOUT output signals change synchronously with the RRSCLKOUT clock signal (in the rising edge of the clock).

The behaviour of this unit is illustrated in the figure below.

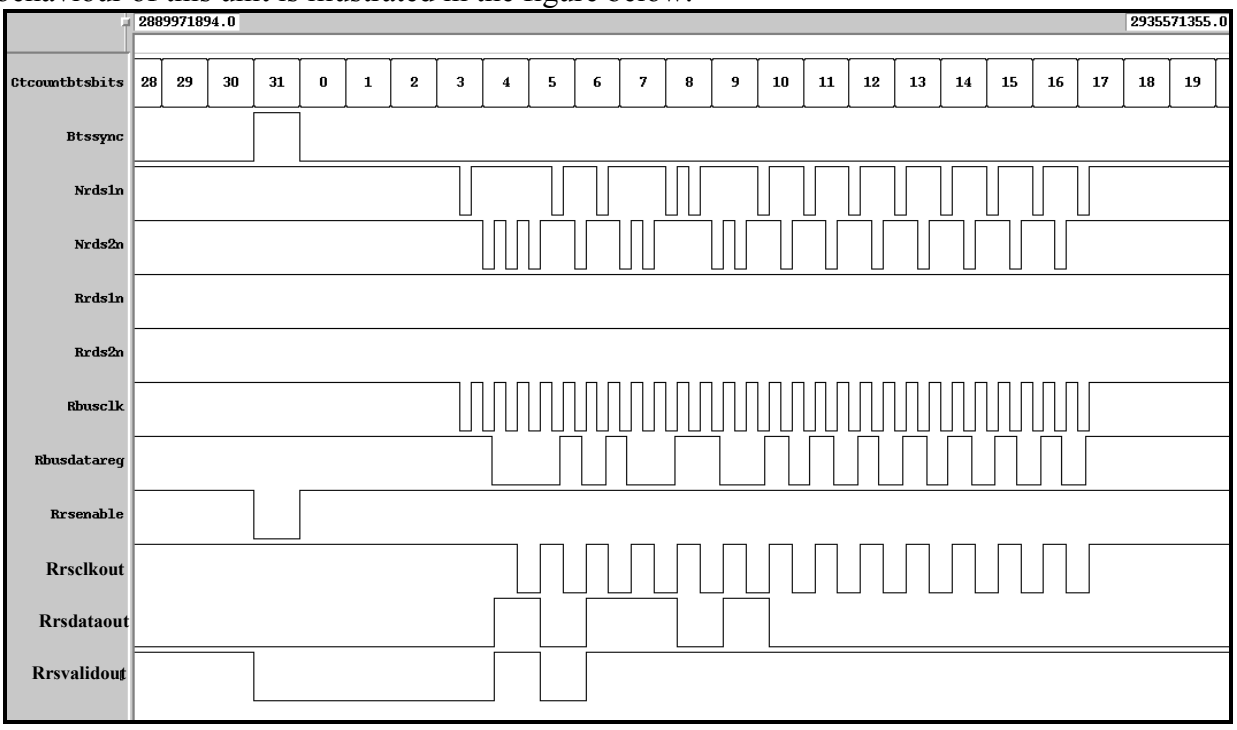


Figure 6.23 RRS Timing.

6.4.6 BTTS Service

CLOCK Signals

- IBusCLK.
- MASTEROSCIN

Input Signals

- OBDHpDHSn
- CTpRTn
- BTTSENABLE
- BTTSDATAIN

Output Signals

- BTTSCLKOUT
- BTTSDATAOUT
- BTTSENABLEReg

This unit implements the behaviour described in AD_2 (page 45 paragraph 6.2.2.7.). Depending on CTpRTn the active clock in this unit is going to be IBusCLK (CTpRTn = '0') or MASTEROSCIN (CTpRTn = '1'). If BTTSENABLE is at high-level logic state, the transmission process starts. BTTSENABLE is then captured every 16 bits, so that just 16-bit blocks of data are transmitted. The BTTSCLKOUT frequency is four times MASTEROSCIN frequency in CT mode and two times IbusCLK in RT mode. The incoming data BTTSDATAIN is sent to the BT-bus transmission module of the BAT2 unit, once it has been correctly captured in BTTSDATAOUT internal signal. BTTSENABLEReg signal enables transmission in the BT-bus.

The behaviour of this unit is illustrated in the figure below.

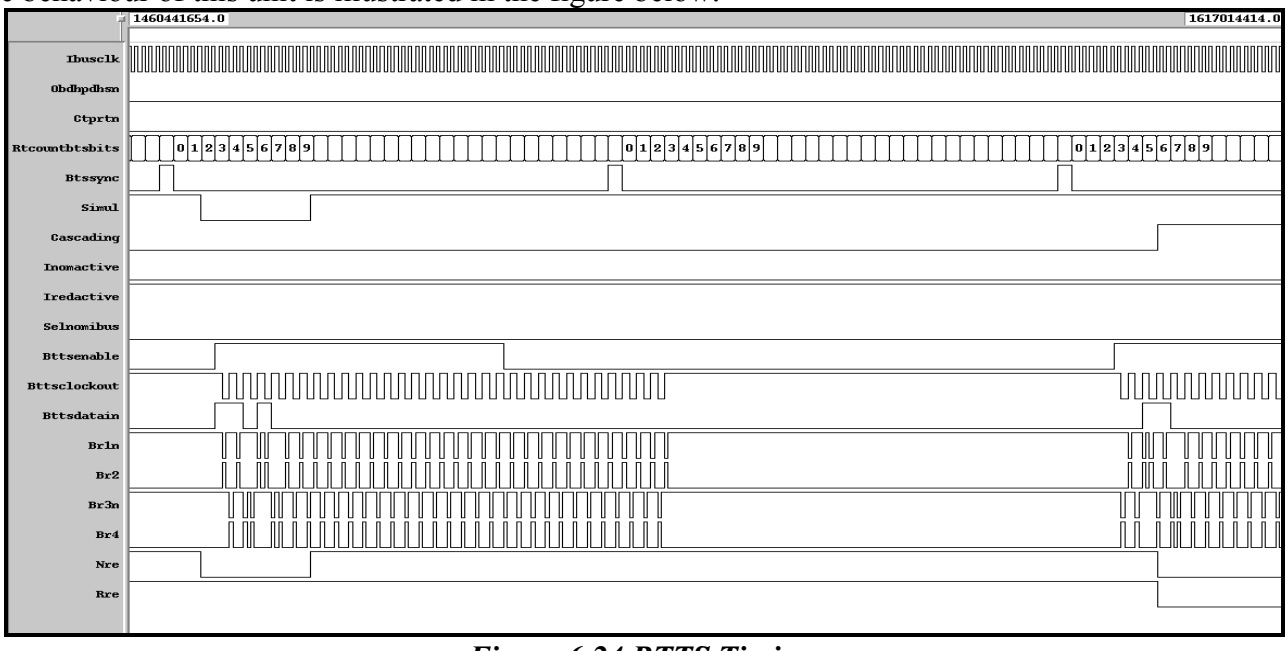


Figure 6.24 BTTS Timing.

6.4.7 BTRS Service

CLOCK Signals

- BTBusCLK.

Input Signals

- OBDHpDHSn
- BTRSENABLEIN
- BTBusDataReg

Output Signals

- BTRSDATAOUT
- BTRSVALIDOUT
- BTRSCLKOUT

This unit implements the behaviour described in AD_2 (page 47 paragraph 6.2.2.8.).
It must be noted that this unit works both in CT and RT modes.

If BTRSENABLEIN is at high-level logic state the reception process start. Data Captured in the Clock and recovering unit (paragraph 7.2.1) is decoded and sent to the BTRSDATAOUT output signal. BTRSVALIDOUT output signal indicates whether a valid or invalid Litton code has been received. BTRSCLOCK is generated by dividing by two the BTBusCLK internal clock.

The behaviour of this unit is illustrated in the figure below.

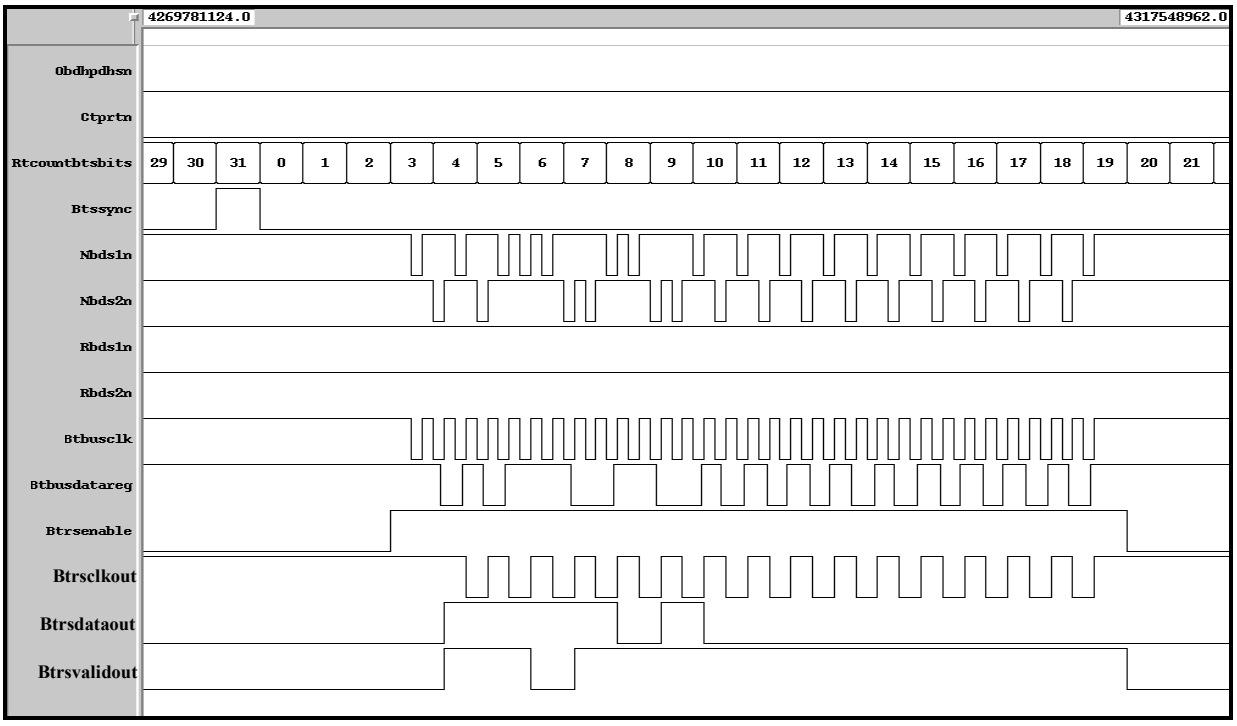


Figure 6.25 BTRS Timing.

6.5 CLOCK DOMAINS

There are six clocks in the DOCC ASIC:

- LOSC
- MASTEROSCIN
- CITMOD
- IBusCLK
- RBusCLK
- BTBusCLK

Although CITMOD and MASTEROSCIN are not going to work never at the same time, because the first one is used in OBDH mode and the second is used in DHS mode. Both share the same pin.

In the block diagram presented in the figure 6.1 every module works with a specific clock, that depends on the mode. The clock domains can be described as follow (see figure 6.26):

- LOSC: The data recovering on the I-Bus, R-Bus and BT-Bus is made with this clock. Also the detection of the synchonization patterns and the activation, deactivation and selection of the set of buses (nominal or redundant) works with the LOSC clock. The signals coming from the buses are captured two times at LOSC rate to avoid metastability and the data modulated in Litton code is decoded into an internal register.
- IBusCLK: This clock is recovered by developing a logic AND of the I-Bus signals NIDS1n and NIDS2n or RIDS1n and RIDS2n. The modules included in the IBusCLK domain are: BAR1/IRS, BAT1/RTS, BAT2/BTTS and IUB when the Asic is connected to a Remote Terminal. The data sent to the Asic trough the I-Bus is decoded two cycles of LOSC after the falling edge of IBusCLK and so (taking into account that the LOSC frequency is much higher) they are stable in the rising edge of IBusCLK. The flow of data between the two clock domains is made safely by this way.
- RBusCLK: This clock is recovered by developing a logic AND of the R-Bus signals NRDS1n and NRDS2n or RRDS1n and RRDS2n. Only the BAR2/RRS module performs its function at RBusCLK rate. The data recovered from the R-Bus are decoded in a register two LOSC cycles after the falling edge of RBusCLK and they can be captured safely in the following rising edge.
- BTBusCLK: This clock is recovered from the BT-Bus by performing a logic AND of the NBDSn1 and NBDSn2 or the RBDS1n and RBDS2n signals. The module BAR3/BTRS works with this clock. The interface between the BTBusCLK domain and the LOSC domain is made as in the others data buses.
- CITMOD/MASTEROSCIN: These clocks are inputs for the Asic. They can share the same pin because CITMOD is used only in OBDH mode and MASTEROSCIN only in DHS mode. The modules that form these clocks domains are BAT1/RTS, BAT2/BTTS and IUB, when they are in CT mode. They belong to the CITMOD domain when the data bus is OBDH and to the MASTEROSCIN domain when it is DHS.

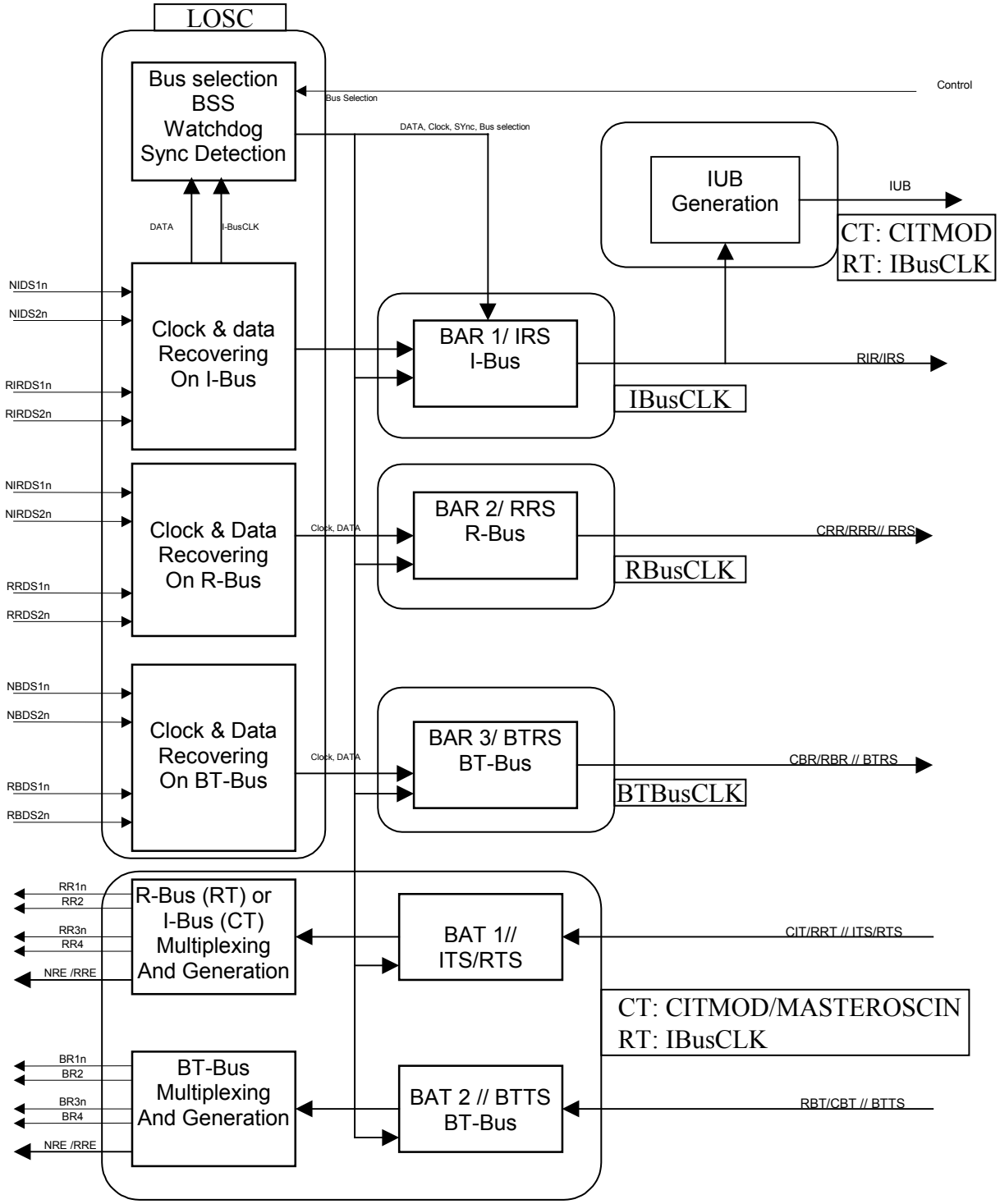


Figure 6.26 Clock Domains

6.6 RESET STRATEGY AND GENERATION OF INTERNAL RESET

In the DOCC Asic there are three internal asynchronous resets. These resets can take different values depending on the working mode of the Asic:

- ResetNInt: This is the general reset, it will force the initial values in almost all the registers of the Asic (except the registers that are initialised with the other two resets, see below). This reset is asserted when the external RESETN is forced to low level, when the Asic is in TESTMODE or when it is in normal mode and it is used as interface with a Control Terminal. When the DOCC Asic interfaces with a Remote Terminal then this general reset acts also when the TIMEOUTn signal is at low level (when both buses are deactivated or when the bus selection changes).

```
If (CTpRTn = '0' and TESTMODE = '0') then
    ResetNInt <= RESETN and TimeOutNInt;
Else
    ResetNInt <= RESETN;
End if;
```

- CRRINITInt: This reset will force the signals CRR/RRR and RRS to their initial values when it is at low level. In TESTMODE and OBDH this internal reset is the negative of the input CRRINIT, so when this input is high the signals CRR/RRR are initialised. In DHS mode CRRINITInt is equal to CRRINIT, so when this input is at low level the RRS signals are initialised.

```
If (OBDHpDHSn = '1' or TESTMODE = '1') then
    CRRINITInt <= not (CRRINIT);
Else
    CRRINITInt <= CRRINIT;
End if;
```

- CBRINITInt: This reset will force the signals CBR/RBR and BTRS to their initial values when it is at low level. In TESTMODE and OBDH this internal reset is the negative of the input CBRINIT, so when this input is high the signals CBR/RBR are initialised. In DHS mode CBRINITInt is equal to CBRINIT, so when this input is at low level the BTRS signals are initialised.

```
If (OBDHpDHSn = '1' or TESTMODE = '1') then
        CBRINITInt <= not (CBRINIT);
Else
        CBRINITInt <= CBRINIT;
End if;
```

6.7 METASTABILITY PROTECTION

The signals that come from the data buses (XXDSIn1 and XXDSIn2) are captured two times with the LOSC clock to avoid metastability. The configuration pins are very stable, so they do not need to be captured. The inputs that comes from the user side are synchronized signals because the Remote Terminal or the Control Terminal have a synchronous design, so they are either dangerous in terms of metastability.

6.8 GLITCH-FREE OUTPUTS DISCUSSION

In this paragraph are presented all the outputs that are generated combinationally and it is going to be done a discussion about the possible existence of glitches in each one. The outputs that don't appear below are generated sequentially and so there can't be glitches in them.

- LOSCDiv4: This output is captured so it can't have glitches.
- TimeOutN: This output depends on the signals: SelNomIbus, SelNomIbusReg, IRedActive and INomActive. All the signals are captured with the LOSC clock except SelNomIbus. This signal depends on CITSEL, CTpRTn, OBDHpDHSn, SELECTINR, INomActive, IRedActive. These signals are all configuration signals (static or quasi-static) or are captured with the LOSC clock, so the TimeOutN output can't have glitches.
- RRn: This bus depends on the configuration signals: OBDHpDHSn and CTpRTn. Depends also on signals captured with the LOSC clock: INomActive, IRedActive and on signals captured with the IbusCLK clock (RRTENReg, RRTDataReg, RTSDATAOUT and RTSENABLEReg) or with the CITMOD clock (RR1Data, EnableITSTransmit, ITSDATAOUT). The signals captured with different clocks don't influence these outputs at the same time, so are not expected glitches by this reason. However some glitches could appear. The logic validation will detect possible glitches in the normal work of the Asic. The connection with the data buses is made through a RC net, and it would filter the glitches that could be produced in these outputs.
- NRE, RRE: These outputs depend on the configuration pins: OBDHpDHSn, CTpRTn, SIMUL and CASCADING, on the quasi-static signal captured with the LOSC clock SelNomIbus and on the captured signals: RRTENReg, RTSENABLESYNC and ITSENABLESYNC. As the signals captured with different clocks are not going to change at the same time, no glitches are expected.
- BRn: This bus depends on the configuration signals: OBDHpDHSn and CTpRTn. Depends also on signals captured with the LOSC clock: INomActive, IRedActive and on signals captured with the IbusCLK clock (RBTENReg, RBTDataReg, RTSDATAOUT and RTSENABLEReg) or with the CITMOD clock (CBTENReg, CBTDataReg, CTBTTSENABLEReg, RTBTTSENABLEReg, CTBTTSDATAOUT, RTBTTSDATAOUT, En2CITMOD). The signals captured with different clocks don't influence these outputs at the same time, so are not expected glitches by this reason. However some glitches could appear. The logic validation will detect possible glitches in the normal work of the Asic. The connection with the data buses is made through a RC net, and it would filter the glitches that could be produced in these outputs.

- NBE, RBE: These outputs depend on the configuration pins: OBDHpDHSn, CTpRTn, SIMUL and CASCADING, on the quasi-static signal captured with the LOSC clock SelNomIbus and on the captured signals: RBTENReg, CTBTTSENABLESYNC and RTBTTSENABLESYNC. As the signals captured with different clocks are not going to change at the same time, no glitches are expected.
- SELECTOUTN: Depends on the static or quasi-static signals: OBDHpDHSn, SIMUL, CASCADING and on the signal SelNomIbus, which is generated through another configuration or captured signals. There can't be glitches in this output.
- SELECTOUTR: Depends on the same signals that SELECTOUTN. It can't have any glitches.
- BTSCLOCKOUT: This signal depends on CTpRTn, CTBTSCLOCKOUTInt and RTBTSCLOCKOUTInt. The first signal comes from a configuration pin and the others two are captured, so no glitches are possible in the BTSCLOCKOUT output.
- BTSSYNC: Depends on CTpRTn, CTBTSSYNC and RTBTSSYNC. The two last signals are captured, there can't be glitches in this output.
- RIRDATA: Depends on OBDHpDHSn, RIRDATAInt and IRSDATAOUT. The first signal is static (configuration signal) and the other two are captured, so there can't be glitches in this output.
- RIRVAL: Depends on OBDHpDHSn and two captured signals: RIRVALInt and IRSVALIDOUT. It has no glitches.
- CRRCLK: Depends on OBDHpDHSn and two captured signals: CRRCLKInt and RRSCLOCKOUT. There are no glitches in this output.
- CRRVAL: Depends on OBDHpDHSn and two captured signals: CRRVALInt and RRSVALIDOUT. There are no glitches in this output.
- CRRDATA: Depends on OBDHpDHSn and two captured signals: CRRDATAInt and RRSDATAOUT. There can't be glitches in this output.
- CBRCLK: Depends on OBDHpDHSn and two captured signals: CBRCLKInt and BTRSCLOCKOUT. There are no glitches in this output.
- CBRVAL: Depends on OBDHpDHSn and two captured signals: CBRVALInt and BTRSVALIDOUT. There are no glitches in this output.
- CBRDATA: Depends on OBDHpDHSn and two captured signals: CBRDATAInt and BTRSDATAOUT. There can't be glitches in this output.
- ITSCLOCKOUT: This output is generates through the signals CTpRTn, ITSCLOCKOUTInt and RTSCLOCKOUT. The first signal is static (configuration input) and the two last are captured, so there can't be glitches in this output.
- BTTSCLOCKOUT: Depends on CTpRTn, and two cptured signals: CTBTTSCLOCKOUTReg and RTNTTSCLOCKOUTReg. This output can't have any glitches.
- CTCLK: The value of this output is generated with the CTpRTn, RTCTCLKInt and CTCTCLKInt signals value. The last two are captured, so no glitches can appear in this output.
- IRCLK: Depends on CTpRTn, and two captured signals: CTIRCLKInt and RTIRCLKInt. No glitches are going to appear in this output.
- TRCLK: Depends on CTpRTn, and two captured signals: CTTRCLKInt and RTTRCLKInt. No glitches are going to appear in this output.

- BCPVAL: Depends on CTpRTn, and two captured signals: CT BCPVAL and RT BCPVAL. No glitches are going to appear in this output.
- BCP: Depends on CTpRTn, and two captured signals: CTBCP and RTBCP. No glitches are going to appear in this output.
- MLADD: Depends on CTpRTn, and two captured signals: CTMLADD and RTMLADD. No glitches are going to appear in this output.
- CHADD: Depends on CTpRTn, and two captured signals: CTCHADD and RTCHADD. No glitches are going to appear in this output.
- PC: Depends on CTpRTn, and two captured signals: CTPC and RTPC. No glitches are going to appear in this output.
- ANCLK: Depends on CTpRTn, and two captured signals: CTANCLKInt and RTANCLKInt. No glitches are going to appear in this output.
- SOC: Depends on CTpRTn, and two captured signals: CTSOC and RTSOC. No glitches are going to appear in this output.
- SH: Depends on CTpRTn, and two captured signals: CTSHInt and RTSHInt. No glitches are going to appear in this output.
- MOSC: Depends on CTpRTn, and two captured signals: CTMOSC and RTMOSC. No glitches are going to appear in this output.
- MOLC: Depends on CTpRTn, and two captured signals: CTMOLC and RTMOLC. No glitches are going to appear in this output.
- MOHL: Depends on CTpRTn, and two captured signals: CTMOHL and RTMOHL. No glitches are going to appear in this output.
- MOBT: Depends on CTpRTn, and two captured signals: CTMOBT and RTMOBT. No glitches are going to appear in this output.
- MODBL: Depends on CTpRTn, and two captured signals: CTMODBL and RTMODBL. No glitches are going to appear in this output.
- MODS16: Depends on CTpRTn, and two captured signals: CTMODS16 and RTMODS16. No glitches are going to appear in this output.
- MODS8: Depends on CTpRTn, and two captured signals: CTMODS8 and RTMODS8. No glitches are going to appear in this output.
- MOANS: Depends on CTpRTn, and two captured signals: CTMOANS and RTMOANS. No glitches are going to appear in this output.
- MOAND: Depends on CTpRTn, and two captured signals: CTMOAND and RTMOAND. No glitches are going to appear in this output.
- DATARRT: Depends on CTpRTn, and two captured signals: CTDATARRTInt and RTDATARRTInt. No glitches are going to appear in this output.
- ENRRT: Depends on CTpRTn, and two captured signals: CTENRRTInt and RTENRRTInt. No glitches are going to appear in this output.

7 TESTS

7.1 FUNCTIONAL VALIDATION

The test plan developed to make the functional validation of this ASIC is fully explained in the "DOCC ASIC Test Plan" document.

Functional Validation at VHDL level shall be defined in order to obtain a good VHDL coverage. The use of VHDLcover will allow measuring Statement, Branch, Condition, and Path coverage. The goal has been 100 % for Statement and Condition, and 80 % for Branch and Path.

7.1.1 Quality of the Testbench

The quality of the set of test performed has been measured by using a tool called VHDLCover (by TransEda) which analysed the quality of a testbench by using metrics applied from software testing ideas. The following items has been measured:

- Statement Coverage.
- Condition Coverage.
- Branch Coverage.
- Path Coverage.

The table bellow presents the different coverage obtained.

	Coverage	**Number Covered**	**Total Number**
STATEMENT	100	879	879
CONDITION	99.6	500	502
BRANCH	96.9	376	388
PATH	79.5	201	253

Table 7.1 Test code quality.

7.2 LOGIC LEVEL VERIFICATION

The previous stimuli have been applied also to the logic level netlist obtained after Place & Route and with the corresponding backanotated delays. The simulation results have been compared against the VHDL RT Level simulation and no differences have been encountered. Three different simulations have been performed:

- Worst Case: WORST Process, 125º C, and 4.5 V.
- Typical Case: TYPICAL Process, 25º C, and 5 V.
- Best Case: BEST Process, -55º C, and 5.5 V.

ANNEXE A IMPLEMENTATION EXAMPLE I: MG2RT.

A.1 FOUNDRY, TECHNOLOGY AND DEVICE SELECTION

One possible example of implementation for this design could be to select the TEMIC Semiconductors foundry; the technology could be the MG2RT 0.5 µm RICMOS IV SOI 5.0V Gate Array family. Specifically, the MG2091E array.

The technical characteristics of the selected device are:

DEVICE: MG2091E	
Total Cells	91,464
Maximum Usable Cells	73,171
Total Pads	235
Maximum Programmable I/Os	212
Die Area (mm2) with scribe line	42.45

Table A.1.1: Technical characteristics for the MG2091E array.

A.2 ELECTRICAL CHARACTERISTICS

A.2.1 Absolute Maximum Ratings

Ambient temperature under bias (TA): Military	-55 to +125 C
Junction temperature	TJ < TA + 20 C
Storage temperature	-65 to +150 C
Supply voltage VDD	-0.5 V to +6 V
I/O voltage	-0.5 V to VDD + 0.5 V

Table A.2.1 Absolute maximum ratings.

A.2.2 DC Characteristics

Specified at VDD = +5 V +/- 10%

Symbol	Parameter	Min	Typ	Max	Unit	Conditions
VIL	Input low voltage	0		0.8	V	
VIH	Input high voltage	2.2		VDD	V	
VOL	Output low voltage			0.4	V	IOL = -12 mA
VOH	Output high voltage	2.4			V	IOH = +12 mA
VT+	Schmitt trigger positive threshold			1.5	V	
VT-	Schmitt trigger negative threshold	1.0			V	
IL	Input leakage No pull up/down Pull up Pull down		 +/-1 -50 114	 +/-5	 µA µA µA	
IOZ	3-State output leakage current		+/-1	+/-5	µA	
IOS	Output short circuit current IOSN IOSP			 48 36	 mA mA	
ICCSB	Leakage current per cell		0.04 0.1 1.0	0.6 2.0 10.0	nA nA nA	Commercial Industrial Military
ICCOP	Operating current per cell		0.45	0.6	µA/MHz	

Table A.2.2 DC characteristics.

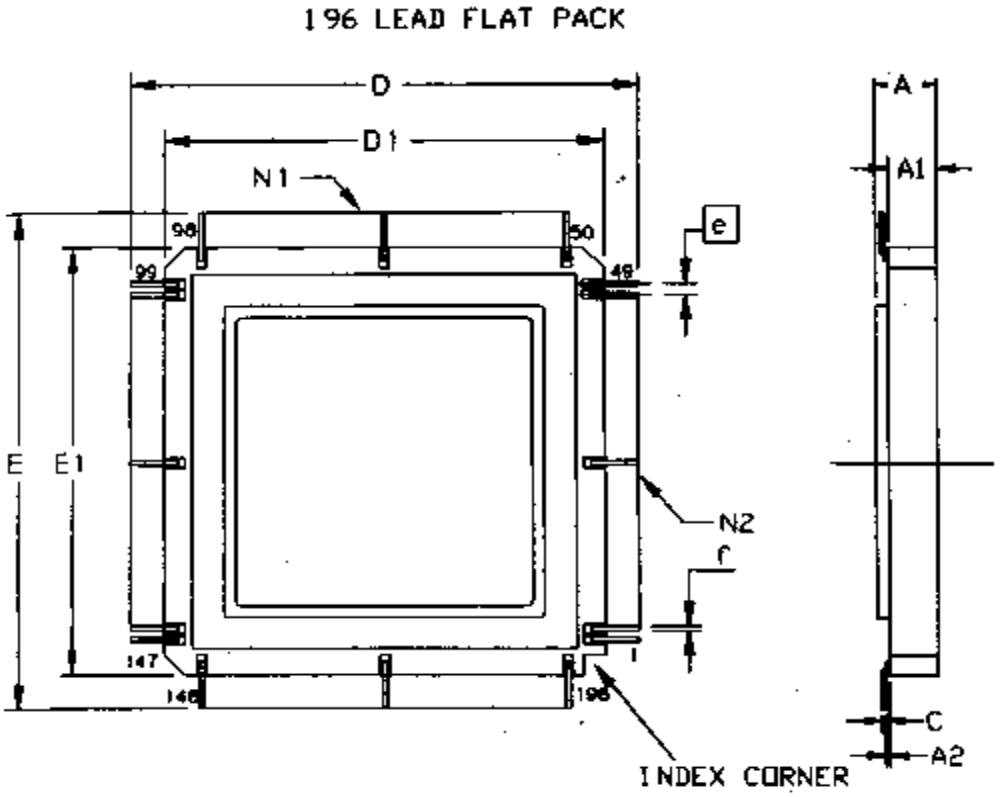
A.2.3 Radiation Characteristics

- Latch up immune.
- Total dose better than:
- 50 Krads when tested to TM1019.4
- 70 Krads when tested to SCC22900 (5V)
- 100 Krads when tested to SCC22900 (3V)
- SEU better than:
- 15 Mev (3V) = 10^{-6} ber for GEO
- 20 MeV (5V) = 10^{-7} ber for GEO
- SEU free cells (better than 100 MeV)

Taking into account the radiation characteristics the possibility of a SEU can be calculated as follows: Number of SEU = 10^{-7} SEU/(Bit * Day) * Number of FF = 10^{-7} * 539 = 5.39 * 10^{-5} SEU/Day. So is expected that appears a SEU each 18553 days.

A.3 PACKAGING

The package that could fit with the DOCC design is the MQFPL196 whose outline is given in Figure A.3.1.



	Min	Max	Min	Max
A	2.13	2.65	.084	.104
A1	1.83	2.24	.072	.088
A2	0.203 REF		.008 REF	
C	0.102	0.203	.004	.008
D/E	46.73	47.94	1.840	1.887
D1/E1	34.03	34.54	1.340	1.360
e	0.635 BSC		.025 BSC	
f	0.20 REF		.008 REF	
L	6.35	6.70	.250	.264
N1	49		49	
N2	49		49	

PACKAGE CODE : U38N U43N U44N U51N
INTERNAL CODE : K9
MHS S.A.
REV : J DATE : 17-01-00

Figure A.3.1 MQFPL196 package.

A.4 GATE COUNT AND POWER CONSUMPTION

A.4.1 Number of Gates

ASIC's core: 7606
Internal scan chain: 1255

Global estimation: **8861**

A.4.2 Power Consumption

The Power consumption of an MG2RT array is due to three factors: leakage (P1), core (P2) and I/O (P3) consumption:

P1+P2+P3 = **24.09 mW**

A.4.2.1 Standby Power Consumption

The consumption due to leakage current may be defined as:

$P1=(VDD-VSS)* I_{CCSB}*N_{CELL}$

The values are shown in the next table:

(VDD-VSS)	= 5 V
I_{CCSB} is the leakage current trough a polarised basic gate	= 1 nA (125°C)
N_{CELL} is the number of used cells	= 8861 gates
P1	**= 0.044 mW**

Table A.4.1 Data to calculate the standby power consumption.

A.4.2.2 Core Power Consumption

The consumption due to the switching of cells in the core of the matrix is bounded by:

$P2= N_{CELL}*P_{GATE}*C_{ACTIVITY}*F$

NCELL is the number of used cells	= 8861 gates
F is the data toggling frequency, is equal to half the clock frequency for data in the worst case	= 8 MHz / 2 = 4 MHz
P_{GATE} is the power consumption per cell	= 1.96 µW/Gate/MHz
$C_{ACTIVITY}$ is the fraction of the total number of cells toggling per cycle	= 20 %
P2	**= 13.89 mW**

Table A.4.2 Data to calculate the core power consumption.

A.4.2.3 I/O Power Consumption

The power consumption due I/O is:

$$P3=Ni*C_O*(VDD-VSS)^2*Fi/2$$

Where:

Ni is equal to the number of output buffers running at Fi (*)	= 65
C_O is the output capacitance	= 50 pF
Fi = F/4	= 1 MHz/4
P3	= **10.16 mW**

Table A.4.3 Data to calculate the I/O power consumption.

It has been considered 1 MHz as the medium data toggling frequency for all the outputs (F = 1 MHz). Also has been considered all the output signals as data with random values (Fi = F/4), although there are some signals that are clocks, but they are only a little bit of them and this simplification will not alter significatively the final value of P3.

A.5 INPUTS/OUTPUTS CHARACTERISTICS

A.5.1 I/O Timing

A.5.1.1 Timing Analysis

The maximum delay at the outputs can be seen below:

```
set_max_delay 1.90941 -to ENRRT
set_max_delay 1.85056 -to DATARRT
set_max_delay 1.85056 -to MOAND
set_max_delay 1.85056 -to MOANS
set_max_delay 1.85056 -to MODS8
set_max_delay 1.85056 -to MODS16
set_max_delay 1.85056 -to MODBL
set_max_delay 1.85056 -to MOBT
set_max_delay 1.85056 -to MOHL
set_max_delay 1.85056 -to MOLC
set_max_delay 1.85056 -to MOSC
set_max_delay 2.63567 -to SH
set_max_delay 2.32799 -to SOC
set_max_delay 2.04529 -to ANCLK
set_max_delay 1.93543 -to PC
set_max_delay 2.32799 -to CHADD[7]
set_max_delay 4.20277 -to CHADD[6]
set_max_delay 2.48575 -to CHADD[5]
set_max_delay 2.32799 -to CHADD[4]
set_max_delay 2.76876 -to CHADD[3]
set_max_delay 2.32799 -to CHADD[2]
set_max_delay 2.32799 -to CHADD[1]
set_max_delay 3.52 -to CHADD[0]
set_max_delay 2.32799 -to MLADD[4]
set_max_delay 2.32799 -to MLADD[3]
set_max_delay 2.45608 -to MLADD[2]
set_max_delay 2.32799 -to MLADD[1]
set_max_delay 2.32799 -to MLADD[0]
set_max_delay 1.93543 -to MLDATA
set_max_delay 1.85056 -to BCP[4]
set_max_delay 1.85056 -to BCP[3]
set_max_delay 1.85056 -to BCP[2]
set_max_delay 1.85056 -to BCP[1]
set_max_delay 1.85056 -to BCPVAL
set_max_delay 2.27529 -to TRCLK
set_max_delay 1.85056 -to IRCLK
```

```
set_max_delay 2.30107 -to CTCLK
set_max_delay 2.88741 -to BTTSCLOCKOUT
set_max_delay 2.53964 -to ITSCLOCKOUT
set_max_delay 2.31993 -to CBRVAL
set_max_delay 1.93543 -to CBRDATA
set_max_delay 2.07697 -to CBRCLK
set_max_delay 2.31993 -to CRRVAL
set_max_delay 1.93543 -to CRRDATA
set_max_delay 2.17212 -to CRRCLK
set_max_delay 1.68684 -to IRSCLOCKOUT
set_max_delay 2.06842 -to RIRVAL
set_max_delay 1.7925 -to RIRSYNC
set_max_delay 1.85056 -to RIRDATA
set_max_delay 2.54207 -to RIRCLK
set_max_delay 2.06823 -to BTSSYNC
set_max_delay 2.92009 -to BTSCLKOUT
set_max_delay 7.85452 -to SELECTOUTR
set_max_delay 6.51477 -to SELECTOUTN
set_max_delay 2.05339 -to ACTIVER
set_max_delay 2.21069 -to ACTIVEN
set_max_delay 9.37254 -to RBE
set_max_delay 7.51921 -to NBE
set_max_delay 9.04514 -to BRn[4]
set_max_delay 8.66743 -to BRn[3]
set_max_delay 9.04514 -to BRn[2]
set_max_delay 8.66743 -to BRn[1]
set_max_delay 9.21123 -to RRE
set_max_delay 7.35789 -to NRE
set_max_delay 9.07278 -to RRn[4]
set_max_delay 8.6948 -to RRn[3]
set_max_delay 9.07278 -to RRn[2]
set_max_delay 8.6948 -to RRn[1]
set_max_delay 6.80968 -to TIMEOUTn
set_max_delay 1.21291 -to LOSCDiv4
```

A.5.1.2 Maximum Operating Frequency

The maximum operating frequency is: **72.9 MHz**

A.5.2 Pinout

A.5.2.1 Power Pins Estimation

It has been selected 12 mA for the output data and 24 mA for the output clocks.

The number of VCC-GND pairs due to the I/O ring is calculated with the formula:
Number of power VCC-GND pairs = (1.5 * Number of inputs + 12 * Number of output data + 24 * Number of output clocks) / 48 = (1.5 * 54 + 12 * 58 + 24 * 12) / 48 ≈ 23

It is necessary to have one power pair more for each 300 flip-flops of the core. As the DOCC Asic has 539 flip-flops there must be two more VCC-GND pairs.

The final number of pins is: Number of I/O + Number of power pins = 124 + 25*2 = **174**

A.5.2.2 I/Os Buffer Selection and Pinout Definition

PIN NUMBER	NAME	TYPE	NOTES
1	LOSCEXTpINTn	BINTTL	5 V
2	OBDHpDHSn	BINTTL	5 V
3	CTpRTn	BINTTL	5 V
4	RESETN	BINTTL	5 V
5	SIMUL	BINTTL	5 V
6	CASCADING	BINTTL	5 V
7	LOSCEXT	BINTTL	5 V
8	LOSCDIV4	BOUT24	24 mA, 5 V
9	VCC		
10	GND		
11	NIDS1n	BINTTL	5 V
12	NIDS2n	BINTTL	5 V
13	RIDS1n	BINTTL	5 V
14	RIDS2n	BINTTL	5 V
15	VCC		
16	GND		
17	NRDS1n	BINTTL	5 V
18	NRDS2n	BINTTL	5 V
19	RRDS1n	BINTTL	5 V
20	RRDS2n	BINTTL	5 V
21	VCC		
22	GND		

23	NBDS1n	BINTTL	5 V
24	NBDS2n	BINTTL	5 V
25	RBDS1n	BINTTL	5 V
26	RBDS2n	BINTTL	5 V
27	VCC		
28	GND		
29	RR1n	BOUT12	12 mA, 5 V
30	RR2	BOUT12	12 mA, 5 V
31	RR3n	BOUT12	12 mA, 5 V
32	RR4	BOUT12	12 mA, 5 V
33	VCC		
34	GND		
35	NRE	BOUT12	12 mA, 5 V
36	RRE	BOUT12	12 mA, 5 V
37	NBE	BOUT12	12 mA, 5 V
38	RBE	BOUT12	12 mA, 5 V
39	VCC		
40	GND		
41	BR1n	BOUT12	12 mA, 5 V
42	BR2	BOUT12	12 mA, 5 V
43	BR3n	BOUT12	12 mA, 5 V
44	BR4	BOUT12	12 mA, 5 V
45	BTSCLKOUT	BOUT24	24 mA, 5 V
46	BTSSYNC	BOUT12	12 mA, 5 V
47	VCC		
48	GND		
49	BTSENABLE	BINTTL	5 V
50	CITMOD/ MASTEROSCIN	BINTTL	5 V
51	ACTIVEN	BOUT12	12 mA, 5 V
52	ACTIVER	BOUT12	12 mA, 5 V
53	VCC		
54	GND		
55	SELECTOUTN	BOUT12	12 mA, 5 V
56	SELECTOUTR	BOUT12	12 mA, 5 V
57	CITSEL/ SELECTINN	BINTTL	5 V
58	SELECTINR	BINTTL	5 V
59	VCC		
60	GND		
61	RRTEN/	BINTTL	5 V

	ITSENABLE/ RTSENABLE		
62	CITDATA/ RRTDATA/ ITSDATAIN/ RTSDATAIN	BINTTL	5 V
63	ITSCLOCKOUT/ RTSCLOCKOUT	BOUT24	24 mA, 5 V
64	IRSENABLE	BINTTL	5 V
65	VCC		
66	GND		
67	RIRDATA/ IRSDATAOUT	BOUT12	12 mA, 5 V
68	RIRVAL/ IRSVALIDOUT	BOUT12	12 mA, 5 V
69	IRSCLOCKOUT	BOUT24	24 mA, 5 V
70	RRRINIT/ CRRINIT/ RRSENABLE	BINTTL	5 V
71	VCC		
72	GND		
73	RRRDATA/ CRRDATA/ RRSDATAOUT	BOUT12	12 mA, 5 V
74	RRRVAL/ CRRVAL/ RRSVALIDOUT	BOUT12	12 mA, 5 V
75	RRRCLK/ CRRCLK/ RRSCLOCKOUT	BOUT24	24 mA, 5 V
76	RBTEN/ CBTEN/ BTTSENABLE	BINTTL	5 V
77	VCC		
78	GND		
79	RBTDATA/ CBTDATA/ BTTSDATAIN	BINTTL	5 V
80	BTTSCLOCKOUT	BOUT24	24 mA, 5 V
81	RBRINIT/ CBRINIT/ BTRSENABLE	BINTTL	5 V

82	RBRDATA/ CBRDATA/ BTRSDATAOUT	BOUT12	12 mA, 5 V
83	VCC		
84	GND		
85	RBRVAL/ CBRVAL/ BTRSVALIDOUT	BOUT12	12 mA, 5 V
86	RBRCLK/ CBRCLK/ BTRSCLOCKOUT	BOUT24	24 mA, 5 V
87	CITCLK	BINTTL	5 V
88	CITSYNC	BINTTL	5 V
89	CTRIRSYNC	BINTTL	5 V
90	CTRIRDATA	BINTTL	5 V
91	CTRIRVAL	BINTTL	5 V
92	VCC		
93	GND		
94	CTRIRCLK	BINTTL	5 V
95	RIRSYNC	BOUT12	12 mA, 5 V
96	RIRCLK	BOUT24	24 mA, 5 V
97	IRCLK	BOUT24	24 mA, 5 V
98	CTCLK	BOUT24	24 mA, 5 V
99	TRCLK	BOUT24	24 mA, 5 V
100	MLDATA	BOUT12	12 mA, 5 V
101	MLADD(0)	BOUT12	12 mA, 5 V
102	VCC		
103	GND		
104	MLADD(1)	BOUT12	12 mA, 5 V
105	MLADD(2)	BOUT12	12 mA, 5 V
106	MLADD(3)	BOUT12	12 mA, 5 V
107	MLADD(4)	BOUT12	12 mA, 5 V
108	VCC		
109	GND		
110	CHADD(0)	BOUT12	12 mA, 5 V
111	CHADD(1)	BOUT12	12 mA, 5 V
112	CHADD(2)	BOUT12	12 mA, 5 V
113	CHADD(3)	BOUT12	12 mA, 5 V
114	VCC		
115	GND		
116	CHADD(4)	BOUT12	12 mA, 5 V
117	CHADD(5)	BOUT12	12 mA, 5 V

118	CHADD(6)	BOUT12	12 mA, 5 V
119	CHADD(7)	BOUT12	12 mA, 5 V
120	VCC		
121	GND		
122	MOSC	BOUT12	12 mA, 5 V
123	MOLC	BOUT12	12 mA, 5 V
124	MOHL	BOUT12	12 mA, 5 V
125	MOBT	BOUT12	12 mA, 5 V
126	VCC		
127	GND		
128	MODBL	BOUT12	12 mA, 5 V
129	MODS16	BOUT12	12 mA, 5 V
130	MODS8	BOUT12	12 mA, 5 V
131	MOANS	BOUT12	12 mA, 5 V
132	MOAND	BOUT12	12 mA, 5 V
133	PC	BOUT12	12 mA, 5 V
134	VCC		
135	GND		
136	ANCLK	BOUT24	24 mA, 5 V
137	SOC	BOUT12	12 mA, 5 V
138	SH	BOUT12	12 mA, 5 V
139	DIGIN	BINTTL	5 V
140	VCC		
141	GND		
142	ANSIN	BINTTL	5 V
143	BCP(1)	BOUT12	12 mA, 5 V
144	BCP(2)	BOUT12	12 mA, 5 V
145	BCP(3)	BOUT12	12 mA, 5 V
146	VCC		
147	GND		
148	BCP(4)	BOUT12	12 mA, 5 V
149	BCPVAL	BOUT12	12 mA, 5 V
150	DATARRT	BOUT12	12 mA, 5 V
151	ENRRT	BOUT12	12 mA, 5 V
152	VCC		
153	GND		
154	TA(0)	BINTTL	5 V
155	TA(1)	BINTTL	5 V
156	TA(2)	BINTTL	5 V
157	TA(3)	BINTTL	5 V
158	VCC		

159	GND		
160	TA(4)	BINTTL	5 V
161	TA(5)	BINTTL	5 V
162	EXTFMT	BINTTL	5 V
163	EXTMLA1	BINTTL	5 V
164	VCC		
165	GND		
166	EXTMLA2	BINTTL	5 V
167	OOINH	BINTTL	5 V
168	TAV	BINTTL	5 V
169	TESTMODE	BINTTL	5 V
170	TESTIBUSCLK	BINTTL	5 V
171	TESTRBUSCLK	BINTTL	5 V
172	TESTBTBUSCLK	BINTTL	5 V
173	SCANENABLE	BINTTL	5 V
174	TIMOUTn	BOUT12	12 mA, 5 V

Table A.5.1 Pinout

A.6 MANUFACTURING TESTS

A.6.1 Manufacturing Test Approach

Manufacturing test shall be performed by using several significant Functional tests plus some tests specifically dedicated to obtain good test coverage. Using SYNOPSYS Test Compiler will automatically generate these tests.

A.6.2 DFT

A full-scan insertion has been used in the ASIC, in order to achieve a high fault coverage result (over the 96%).

In the full-scan design technique, all sequential cells in the design are modified to perform a serial shift function. Sequential elements that are not scanned are treated as black box cells (cells with unknown function).

Full-scan divides a sequential design into combinational blocks as shown in figure A.6.1. Ovals represent combinational logic; rectangles represent sequential logic. The full-scan diagram shows the scan path through the design.

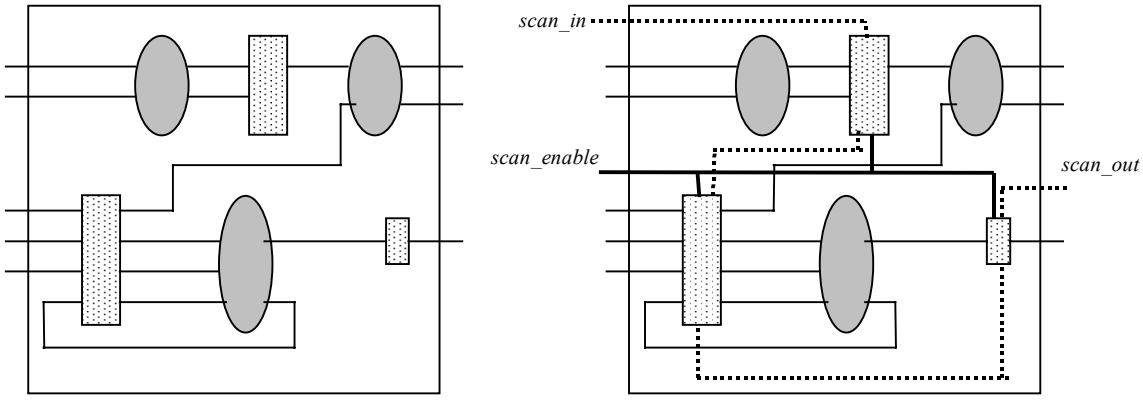


Figure A.6.1 Scan Path through a Full-Scan Design.

Through pseudo-primary inputs, the scan path enables direct control of inputs to all combinational blocks. Through pseudo-primary outputs, the scan path enables direct observability of outputs from all combinational blocks. Efficient combinational ATPG algorithms could be used to achieve high fault coverage results on the full-scan design.

TEST COMPILER (Synopsys) has been used as full-scan insertion and ATPG tool.

This technique usually produces a large number of vectors for the fabrication test. In order to reduce the number of vectors AEO will introduce as much scan chains as possible. The number of scan chains will be defined according to the number of available IOs in the selected package, and the number of test vectors produced. We have selected a number of 15 scan chains. Those chains are as balanced as possible and they are in different clock domain. All the registers in a chain change its values with the same clock.

Scan inputs and outputs will be shared with another signals of the design, in order to no increment the total pin number of the ASIC. In the next table are displayed the inputs and outputs of the 15 scan chains (with the ports of the design with they coincide) and their length.

Scan Chain Number	Design Input	Design Output	Length (Cells number)
1	TA[0]	MLADD[0]	49
2	TA[1]	MLADD[1]	2
3	TA[2]	MLADD[2]	57
4	TA[3]	MLADD[3]	46
5	TA[4]	MLADD[4]	9
6	TA[5]	CHADD[0]	21
7	EXTMLA[1]	CHADD[1]	8
8	EXTMLA[2]	CHADD[2]	2
9	EXTFM	CHADD[3]	43
10	OOINH	CHADD[4]	49
11	TAV	CHADD[5]	56
12	CASCADING	CHADD[6]	48
13	SIMUL	CHADD[7]	45
14	DIGIN	SOC	56
15	ANSIN	SH	48

Table A.6.1 Scan chains inputs, outputs and length.

The scan chain area is 1255 gates and the ASIC's core has 7606 gates, so the scan test increase the ASIC area in 14.16 %. The DOCC ASIC has a total number of 539 flip-flops. All of them are in a scan chain.

The test has been made with a clock period of 500 ns. The following graphic shows the timings for all the ASIC clocks and the moment in which the input stimuli are introduced and the outputs are taken. Those waveforms have been designed to assure the correct function of the test.

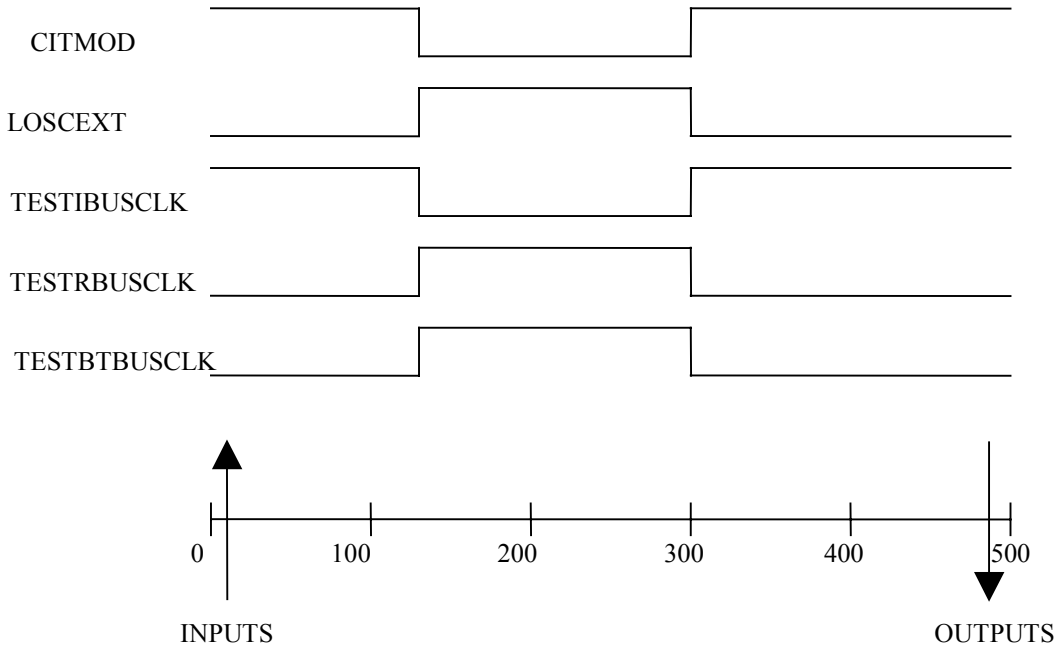


Figure A.6.2 Clocks waveforms and strobes.

A.6.3 Fault Coverage

The final fault coverage is **97.33%**. It is necessary **283** test vectors to assure the coverage. Taking the last data into account the total number of cycles to run the test is:

Number of cycles = Test Vectors Number * Max. Length of the chains = 283 * 57 = **16131**.

A.6.4 DC Parameters Measure

The DC parameters that will be measured during test are: VIH and VIL for all the inputs and VOH and VOL for all the outputs.

A.6.3 AC Parameters Measure

There is a limit in the number of AC parameters that can be measured in the test. It can be done only 15 measures. The measures selected for the DOCC Asic are presented in the Table A.6.3. The first two columns contain the reference signal and edge (rising or falling) respect to which the delay is going to be calculated, the third column indicates the name of the signal whose delay is measured and the last column is the working mode of the Asic.

Input Signal		Output Signal	Mode
Name	Edge	Name	
TESTIBUSCLK	\	RIRCLK	OBDH-RT
TESTRBUSCLK	\	CRRCLK	OBDH-CT
TESTIBUSCLK	/	IRSCLOCKOUT	DHS-RT
TESTIBUSCLK	/	RTSCLOCKOUT	DHS-RT
CITMOD	/	BTTSCLOCKOUT	DHS-CT
SIMUL	/	NRE	OBDH-RT
SIMUL	/	RBE	DHS-CT
CITMOD	/	RR1n	OBDH-CT
TESTIBUSCLK	/	RR2	DHS-RT
CITMOD	/	BR3n	DHS-CT
TESTIBUSCLK	\	MLADD(3)	OBDH-RT
TESTIBUSCLK	\	BCP(2)	DHS-RT
CITMOD	\	MOLC	OBDH-CT
TESTIBUSCLK	/	IRSDATAOUT	DHS-RT
TESTIBUSCLK	\	RRRDATA	OBDH-RT

Table A.6.2 AC parameters to measure.

ANNEXE B IMPLEMENTATION EXAMPLE II: ACTEL FPGA.

B.1 FOUNDRY, TECHNOLOGY AND DEVICE SELECTION

One possible example of implementation for the DOCC IP could be to select an ACTEL FPGA. Taking into account the number of pins, the area, the number of clocks, etc, this design could be implemented in a RT54SX72S FPGA.

The technical characteristics of the selected device are:

DEVICE: RT54SX72S	
Capacity	
Typical Gates	72,000
System Gates	108,000
Logic Modules	6,036
Combinatorial Cells	4,024
SEU Hardened Register Cells	2,012
Maximum Flip-Flops	4,024
Maximun User I/Os	212
Clocks	3
Quadrant Clocks	4
Clock-to-Out	11.0 ns
Input Set-Up(External)	-3.3 ns
Speed Grades	Std, -1
Avalaible Package (CQFP)	208, 256

Table B.1.1: Technical characteristics for the RT54SX72S FPGA.

B.2 ELECTRICAL CHARACTERISTICS

B.2.1 Maximum Operating Conditions.

The absolute maximum ratings for the RT54SX72S FPGAs are described bellow. This information has been extracted from RD_01 reference.

Electrical Specifications	
V_{CCI} (DC Supply Voltage)	-0.3 to +6.0 Volts
V_{CCA} (DC Supply Voltage)	-0.3 to +3.0 Volts
V_I (Input Voltage)	-0.5 to +5.5 Volts
V_O (Output Voltage)	-0.5 to +V_{CCI}+0.5 Volts
Radiation Hard Specifications	
Total Dose- rad(Si)	100 Krad
SEU- errors/bit-day	< 1 E-10
Latchup	Immune
Temperature Specifications	
T_{stg} (Storage Temperature)	- 65 to +150 ºC
Maximum Power Dissipation	1.14 W

Table B.2.1 Maximum operating conditions for the RT54SX72S FPGA.

Note: Maximum Power Dissipation Calculation.

$$Power = \frac{Max.\ Junction\ temp.\ (ºC) - Max.\ Military\ temp.\ (ºC)}{\theta_{ja}(ºC/W)} = \frac{150\ ºC - 125\ ºC}{22\ ºC/W} = 1.14\ W$$

B.2.2 Recommended Operating Conditions.

The Recommended Operating Conditions for RT54SX72S FPGAs are described bellow. This information has been extracted from RD_01 reference.

Temperature Range	- 55 to + 125 ºC
Power Supply Tolerance	Vcc ± 10% Volts.

Table B.2.2: Recommended operating conditions for the RT54SX72S FPGA.

B.2.3 Electrical Characteristics "DC".

The main electrical characteristics for the RT54SX72S FPGA family are described in the next paragraph:

V_{OH}	I_{OH} = -20 µA (CMOS)	V_{CCI} -0.1 < V_{OH} < V_{CCI} Volts
	I_{OH} = -8 mA (TTL)	
	I_{OH} = -6 mA (TTL)	2.4 < V_{OH} < V_{CCI} Volts
V_{OL}	I_{OL} = 20 µA (CMOS)	
	I_{OL} = 12 mA (TTL)	
	I_{OL} = 8 mA (TTL)	< 0.50 Volts
V_{IL}		0.8 Volts
V_{IH}		> 2.0 Volts
C_{IO} (I/O Capacitance)		< 10 pF
I_{CC} (Standby Current)		< 25 mA

Table B.2.3: Electrical characteristics for the RT54SX72S FPGA.

B.2.4 Electrical Characteristics "AC".

The DOCC FPGA is going to operate with five clocks: LOSC (f = 8 Mhz), CITMOD (2*Fbit in OBDH mode and 4*Fbit in DHS mode), IbusClk (Fbit), RbusClk(Fbit) and BTbusClk (Fbit), where Fbit = 524488 Hz. In order to avoid metastability problems, the Input signals have been captured twice in the FPGA in those critical lines, when it was possible.

B.3 PACKAGING

The package that could fit with the DOCC design is the CQFP 208-pin.

B.4 GATE COUNT AND POWER CONSUMPTION

B.4.1 Gate Count.

Number of ports: 124
Number of nets: 1941
Number of cells: 1834
Number of references: 91

Combinational area: 1344
Noncombinational area: 541

Total cell area: 1885

B.4.2 Power Estimation.

FPGA Data Book presents the next expression in order to estimate the Power consumption:

$$P=[I_{CC}standby+I_{CC}active]*V_{CCA}+I_{OL}*V_{OL}*N+I_{OH}*(V_{CCA}-V_{OH})*M$$

Where:

I_{CC} standby is the current flowing when no inputs or outputs are changing.
I_{CC} active is the current flowing due to CMOS switching.
I_{OL}, I_{OH} are TTL sink/source currents.
V_{OL}, V_{OH} are TTL level output voltages.
N equals the number of outputs driving TTL loads to V_{OL}.
M equals the number of outputs driving TTL loads to V_{OH}.

The static power component is typically a small component of the overall power. For military, worst case conditions a typical value is:

P_{static} = 68 mW

The active Power component can also be calculated by means of the following expression:

$$P_{active} = V_{CCA}{}^2 * \begin{bmatrix} (m * C_{EQM} f_m)_{modules} + (n * C_{EQI} * f_n)_{inputs} + \\ 0.5 * (q_1 * C_{EQCR} * f_{q1})_{RCLKA} + (r_1 * f_{q1})_{RCLKA} + \\ 0.5 * (q_2 * C_{EQCR} * f_{q2})_{RCLKB} + (r_2 * f_{q2})_{RCLKB} + \\ 0.5 * (s_1 * C_{EQHV} * f_{s_1})_{HCLK} + (C_{EQHF} * f_{s_1})_{HCLK} \end{bmatrix} + V_{CCI}{}^2 * \left[(p * (C_{EQO} + C_L) * f_p)_{outputs} \right]$$

Where Power is in microwats, Voltage is in Volts, Capacity is in pF, and frequency is in Mhz. The following items form the previous expression:

m	=	Number of logic modules switching at fm.
n	=	Number of input buffers switching at fn.
p	=	Number of output buffers switching at fp.
q_1	=	Number of clock loads on the first routed array clock.
q_2	=	Number of clock loads on the second routed array clock.
r_1	=	Fixed capacitance due to first routed array clock.
r_2	=	Fixed capacitance due to second routed array clock.
s_1	=	Fixed number of clock loads on the dedicated array clock.
C_{EQM}	=	Equivalent capacitance of logic modules in pF.
C_{EQI}	=	Equivalent capacitance of input Buffers in pF.
C_{EQO}	=	Equivalent capacitance of output Buffers in pF.
C_{EQCR}	=	Equivalent capacitance of routed Array Clock in pF.
C_{EQHV}	=	Variable capacitance of dedicated Array Clock in pF.
C_{EQHF}	=	Fixed capacitance of dedicated Array Clock in pF.
C_L	=	Output load capacitance in pF.
f_m	=	Average buffer switching rate in MHz.
f_p	=	Average logic module switching rate in MHz.
f_n	=	Average input output buffer switching rate in MHz.
f_{q1}	=	Average first routed array clock rate in MHz.
f_{q2}	=	Average second routed array clock rate in MHz.
f_{s1}	=	Average second dedicated array clock rate in MHz.

The following values can be used:

m	=	80% of modules. = 1885 * 0.8 = 1508
n	=	inputs/4 = 54/4 = 13.5
p	=	outputs /4 = 70/4 = 17.5
q_1	=	40% of sequential modules = 541 * 0.4 = 216.4
q_2	=	40% of sequential modules = 541 * 0.4 = 216.4
r_1	=	194 pF
r_2	=	194 pF
s_1	=	1080

C_{EQM} = 2.0 pF
C_{EQI} = 1.4 pF
C_{EQO} = 7.4 pF
C_{EQCR} = 1.9 pF
C_{EQHV} = 1.2 pF
C_{EQHF} = 195 pF
C_L = 35 pF.
F = 8 MHz (Clock frequency)
f_m = F/10 = 8 MHz / 10 = 0.8 MHz.
f_n = F/5 = 8 MHz / 5 = 1.6 MHz.
f_p = F/10 = 8 MHz / 10 = 0.8 MHz.
f_{q1} = F/2 = 8 MHz / 2 = 4 MHz.
f_{q2} = F/2 = 8 MHz / 2 = 4 MHz.
f_{s1} = F = 8 MHz.

Thus:

$m * C_{EQM} * f_m = 1508 * 2.0 * 0.8 = 2412.8$
$n * C_{EQI} * f_n = 13.5 * 1.4 * 1.6 = 30.24$
$p * (C_{EQO} + C_L) * f_p = 17.5 * (7.4 + 35) * 0.8 = 593.6$
$0.5 *(q_1 * C_{EQCR} * f_{q1}) = 0.5 * 216.4 * 1.9 * 4 = 822.32$
$r_1 * f_{q1} = 194 * 4 = 776$
$0.5 *(q_2 * C_{EQCR} * f_{q2}) = 0.5 * 216.4 * 1.9 * 4 = 822.32$
$r_2 * f_{q2} = 194 * 4 = 776$
$0.5 *(s_1 * C_{EQHV} * f_{s1}) = 0.5 * 1080 * 1.2 * 8 = 5184$
$C_{EQHF} * f_{s1} = 195 * 8 = 1560$

Power Active = $2.7^2 * (2412.8 + 30.24 + 822.32 + 776 + 822.32 + 776 + 5184 + 1560) + 5.5^2 * (593.6) = 2.7^2 * 12383.68 + 5.5^2 * 593.6 = 108.23$ mW

Power = Power static + Power Active = 68 + 108.23 = 176.23 mW

Power = **176.23 mW**

B.5 INPUTS/OUTPUTS TIMING

The Following table shows the main circuit timing characteristics.

	WORST (125º, 4.5 V)	TYP (25º, 5V)	BEST (-55º, 5.5)
INPAD to GATED	146.4	105.8	46.7
INPAD to CLK (Clock Skew)	95.2	68.7	30.4
INPAD to ASYNC (Reset Delay)	100.5	72.5	32.1
INPAD to OUTPAD	117.5	84.7	37.5
CLOCK to OUTPAD	91.0	65.7	29.1
CLK to CLK			
LOSC	64.1	46.4	20.5
TESTIBUSCLK	57.4	41.7	18.3
TESTRBUSCLK	11.3	8.2	3.5
TESTBTBUSCLK	12.0	8.8	3.7
CITMOD/MASTEROSCIN	52.0	37.8	16.7

Table B.5.1 Timing characteristics for the DOCC FPGA.

The following tables present the setup and hold times for the inputs of the FPGA in the three operating conditions analysed (i.e. Max, Min, Typ).

		WORST (125º, 4.5 V)		TYP (25º, 5 V)		BEST (-55º, 5.5 V)	
		Setup time	Hold time	Setup time	Hold time	Setup time	Hold time
ANSIN	CITMOD	-14.6	16.8	3.4	-1.8	1.5	-0.9
BTSENABLE	CITMOD	11.4	-7.4	8.1	-4.6	3.6	-2.1
CITCLK	CITMOD	19.9	-12.1	14.3	-8.6	6.3	-4.1
CITDATA	CITMOD	15.4	-6.0	11.1	-4.2	5.0	-2.0
CITSEL	TESTIBUSCLK	84.9	-25.1	63.9	-19.4	27.2	-8.2
CITSYNC	CITMOD	18.2	-11.2	13.0	-7.9	5.8	-3.7
CTPRTN	TESTIBUSCLK	116.9	-46.7	84.5	-33.8	38.4	-16.4
CTRIRCLK	CITMOD	-1.0	2.2	-0.7	1.6	-0.3	0.7
CTRIRDATA	CITMOD	-1.1	2.1	-0.8	1.6	-0.3	0.7
CTRIRSYNC	CITMOD	-0.6	1.7	-0.4	1.3	-0.1	0.6
CTRIRVAL	CITMOD	-1.1	2.2	-0.8	1.7	-0.2	0.6
DIGIN	TESTIBUSCLK	-7.2	9.5	-5.0	6.7	-2.2	3.0
EXTFMT	TESTIBUSCLK	22.3	-8.3	16.1	-6.1	7.1	-3.0
EXTMLA_1_	TESTIBUSCLK	24.9	-12.5	18.0	-9.1	8.1	-4.3
EXTMLA_2_	TESTIBUSCLK	34.6	-8.8	24.9	-6.4	11.1	-3.2
IRSENABLE	TESTIBUSCLK	-12.8	17.0	-9.3	12.3	-4.2	5.5
LOSCEXT	TESTIBUSCLK	110.3	-49.8	79.8	-35.9	35.6	-17.1

LOSCINT	TESTIBUSCLK	111.2	-50.7	80.4	-36.5	35.9	-17.4
NBDSN_1_	TESTBTBUSCLK	19.3	-12.4	14.0	-8.9	6.3	-4.6
NBDSN_2_	TESTBTBUSCLK	20.2	-13.1	14.7	-9.5	6.3	-4.1
NIDSN_1_	TESTIBUSCLK	72.4	-16.0	52.4	-11.3	22.1	-4.7
NIDSN_2_	TESTIBUSCLK	68.3	-13.0	51.9	-10.6	22.1	-5.4
NRDSN_1_	TESTRBUSCLK	16.5	-14.9	12.0	-10.9	5.3	-3.9
NRDSN_2_	TESTRBUSCLK	18.9	-17.2	13.8	-12.6	5.9	-5.5
OBDHPDHSN	TESTIBUSCLK	105.6	-45.9	78.8	-34.3	34.7	-15.2
OOINH	CITMOD	25.0	-22.2	18.1	-15.9	8.0	-7.1
RBDSN_1_	TESTBTBUSCLK	19.4	-12.5	14.1	-9.0	6.0	-4.5
RBDSN_2_	TESTBTBUSCLK	18.8	-11.4	13.7	-8.2	5.9	-3.6
RBTDATA	CITMOD	3.0	0.0	-7.6	9.4	1.1	0.0
RBTEN	CITMOD	2.4	-0.5	1.8	-0.2	0.9	-0.2
RIDSN_1_	TESTIBUSCLK	68.5	-13.2	49.6	-9.6	22.2	-5.5
RIDSN_2_	TESTIBUSCLK	67.8	-12.2	49.2	-8.9	22.8	-5.4
RRDSN_1_	TESTRBUSCLK	16.7	-15.1	12.2	-11.1	5.2	-4.8
RRDSN_2_	TESTRBUSCLK	17.6	-15.5	12.8	-11.3	5.6	-5.0
RRTEN	TESTIBUSCLK	-4.8	9.2	-3.3	6.8	-1.6	2.9
SELECTINR	TESTIBUSCLK	81.7	-24.6	61.5	-19.0	26.2	-8.1
TAV	TESTIBUSCLK	38.3	-20.7	27.8	-14.8	11.8	-2.9
TA_0_	TESTIBUSCLK	20.4	-15.5	14.7	-11.0	6.6	-5.1
TA_1_	TESTIBUSCLK	20.7	-16.3	15.1	-11.7	6.7	-5.3
TA_2_	TESTIBUSCLK	16.9	-13.2	12.2	-9.6	5.4	-4.4
TA_3_	TESTIBUSCLK	17.7	-13.1	12.7	-9.4	5.7	-4.3
TA_4_	TESTIBUSCLK	16.7	-13.2	12.0	-9.4	5.4	-4.4
TA_5_	TESTIBUSCLK	14.7	-8.4	10.6	-6.0	4.7	-2.9
TESTMODE	TESTIBUSCLK	63.6	-5.5	43.9	-2.7	20.3	-1.5

Table B.5.2 Setup and Hold times for the DOCC FPGA.