ESA R&D n° 20167/06/NL/FM

Further Development of the Spacecraft Controller on a Chip

AHBR module
Specification and architecture

	Name and Function	Date	Signature
Prepared by	**Florent MERLAY** *Digital design engineer*	2009/03/03	
Verified by	**Marc SOUYRI** *ASIC expert*	2009/03/03	
Authorized by	**Franck KOEBEL** *SCOC3 Asic Responsible*	2009/03/03	

Document type	Nb WBS	Keywords

DOCUMENT CHANGE LOG

Issue/ Revision	Date	Modification Nb	Modified pages	Observations
0/0	23/11/06			First version
0/1	16/16/07			Update with new architecture
0/2	05/03/07			New synthesis result after cod modification
0/3				
0/4	07/11/07			Added : - APB registers description - Interrupt generation - HLOCK management
0/5	09/06/08			Minor updates
	25/08/08		8, 12	5.2: IOInhib management modified * single area reachable when set. * depth defined using EEPROMSIZE Clock definition compliant wit SCOC3 implementation
0/5b	13/12/2013			Updates for ESA IP release: - Gate count update (section 7) - IOINHIB management modified: range protection [plow;phigh] (requirement 5090, pin list)

PAGE ISSUE RECORD

Issue of this document comprises the following pages at the issue shown

Page	Issue/ Rev.	Page	Issue/ Rev.	Page	Issue/ Rev.	Page	Issue/ Rev.	Page	Issue/ Rev.	Page	Issue/ Rev.
all	00/04										

TABLE OF CONTENTS

1 INTRODUCTION

This document is the specification of the AHBR module. This module is part of the SCoC3 design. This document is written in the frame of the ESA R&D "Further development of the Spacecraft Controller on a Chip" reference 20167/06/NL/FM.

Requirements are specified throughout this document in table format as follows:

Id	Requirement Text	Verification Method	Upper Links

- the absolute requirement identifier (Id), defined on 4 digits. The first digit corresponds to the current chapter number.
- the requirement text. If tables are considered as part of requirement they are referenced clearly in the text and inserted after and separated from the requirement table and are managed as free text attached to the identifier requirement.
- the verification method (Verif method) as one of the following : R/review of design; I/inspection; A/analysis; T/test, S/similarity, D/definition (not be tracked).
- the trace to the upper level requirements (Upper Links), shall be managed with the following format:
  - **AAAANNNN** where AAAA is a label associated to the upper document and NNNN the requirement identifier of this upper level.
  - Or **CREATED** key word if the requirement has no link with upper level

All document elements, which are not presented in the table format explained above are not requirements and will not be verified or tracked.

The AHBR module shall perform the connection between two AHB Buses clock synchronous clocks with defined frequency ratio function.

The document contains the following sections :

- Applicable and reference documents followed by the list of the acronyms
- A general description of the module in its environment, and of its content
- A detailed description of the module functionality. This section starts by the description of the operating modes of the module. Then all the functions are described
- A description of the interfaces
- The timing characteristics
- Miscellaneous requirements such as testability, cares for routing etc….

2 DOCUMENTS AND ACRONYMS

2.1 APPLICABLE DOCUMENTS

AD-1	ECSS Q60-02-A: ASIC/FPGA Development Standard, 2007-07-17
AD-2:	VHDL Modelling Guidelines, ftp://ftp.estec.esa.nl/pub/vhdl/doc/ModelGuide.pdf
AD-3	ESA R&D " Further Development of the Spacecraft Controller on a Chip ".Statement o Work
AD-4	Advanced Microcontroller Bus Architecture (AMBA™) Specification, revision 2.0, ARM IHI 0011A

2.2 REFERENCE DOCUMENTS

RD-1	Technical Documentation from Call-Off Order #3 (Spacecraft Controller On-a Chip) o ESA contract #13345/99/NL/FM (Building Blocks for System On-a Chip), known t both parties
RD-2:	Contract #13345/99/NL/FM (Building Blocks for System On-a Chip)
RD-3	Synthesisable IP cores available from ESA
RD-4	ASIC Design and Manufacturing Requirements, ESA document WDN/PS/700
RD-5	Minutes of informal meeting R&D.SOC.MN.00395.V.ASTR from 15. December 2005, known to both parties

2.3 ACRONYMS

AD	Applicable Document
ADR	Architectural Design Review
AIT	Assembly Integration and Test
AHB	AMBA High speed Bus
AMBA	Advanced Microcontroller Bus Architecture
ASIC	Application Specific Integrated Circuit
ASIM	Application Specific Integrated Microsystem
ASSP	Application Specific Standard Product
BLADE	Board for LEON and Avionics DEvelopment
CDR	Critical Design Review
CPU	Central Processor Unit
DDR	Detailed Design Review
DFF	D-Type Flip Flop
DRC	Design Rule Check
DSP	Digital Signal Processor
EDAC	Error Detection And Correction
EDA	Electronic Design Automation
EGSE	Electrical Ground Support Equipment
ESA	European Space Agency
ESTEC	European Space Research and Technology Centre
FDIR	Failure Detection Isolation and Recovery
FPGA	Field Programmable Gate Array
GEO	Geosynchronous Equatorial Orbit
GRLIB	Gaisler Research Library,
HDL	Hardware Description Language
I/O	Input/Output
ID	Identification
IDR	Initial Design Review
IEEE	Institute of Electrical and Electronics Engineers
IP, IPR	Intellectual Property, Intellectual Property Rights
IPMON	Performance Monitoring (IP block)
ITT	Invitation To Tender
JTAG	Joint Test Action Group (refer to IEEE std 1149.1)
LEO	Low Earth Orbit
LET	Linear Energy Transfer
OBDH	On Board Data Handling
OBMU	On Board Management Unit
PCB	Printed Circuit Board
PDF	Portable Document Format
PDR	Preliminary Design Review
PM	Performance Monitoring (in fact called IPMON)
PM	Processor Module
RD	Reference Document
RTEMS	
RTOS	Real Time Operating Syster (example: RTEMS)
SOC	System On a Chip
SCoC	Spacecraft Controller on a Chip
SEE	Single Event Effect (or SEP Single Event Phenomena)
SEL	Single Event Latch up
SEP	see SEE
SET	Single Event Transient
SEU	Single Event Upset
SRAM	Static Random Access Memory
SRR	Specification Requirement Review
TC	TeleCommand
TID	Total Integrated Dose
TM	TeleMetry
TRP	Technological Research Programme
VHDL	VHSIC Hardware Description Language,
VLSI	Very Large Scale Integratio
WP	Work Package
WWW	World Wide Web

3 GENERAL DESCRIPTION OF THE MODULE

The AHB to AHB Bridge (AHBR) module is intended to be used to connect the CPU AHB Bus and the IO AHB Bus on SCOC3.

The AHBR is composed of the following functions (Figure 1):

- AHB Slave interface
- AHB Master interface
- Interconnection block between AHB Slave interface and AHB Master interface

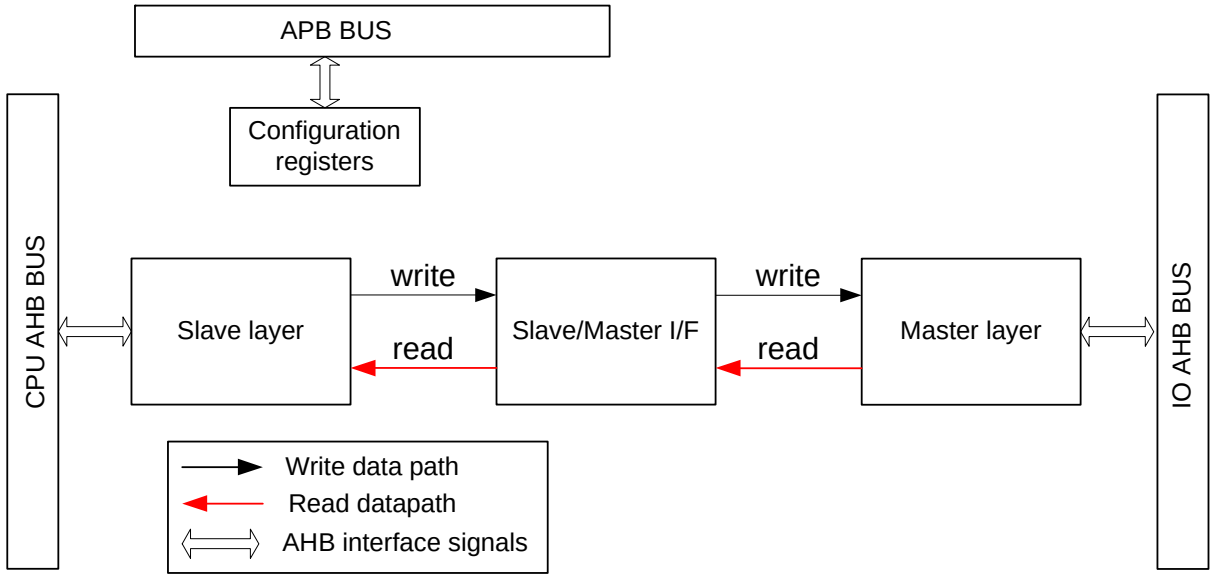

Figure 1 – AHBR block diagram description

4 GENERAL REQUIREMENTS

4000	AHB to AHB Bridge shall be compliant AMBA 2.0		

4010	Single and Burst AHB transfers (Burst 2, 3 or 4) shall be supported.		

4020	Burst WRAP mode shall not be supported.		

4030	The bridge shall only be able to perform CPU to IO accesses.		

4040	The bridge shall manage different clock frequency ratio between slave side an master side by a configuration input port : 1x, 2x, 3x and 4x		
4050	Internal registers shall be used for data buffering		
4060	Different transfer size shall be supported byte, half word and word.		
4070	OKAY and ERROR responses shall be supported		
4080	Posted write shall be used for all access types : SINGLE or BURST.		
4090	AHB slave interface shall manage all slave signals required on an AMBA AHB Bus		
4100	AHB master interface shall manage all master signals required on an AMBA AH Bus.		
4110	Deleted		
4120	Endianness shall be Big Endian.		
4130	Early burst termination shall not be supported		
4140	The bridge shall perform locked accesses on the IO AHB bus when locked accesse are being requested to it on the CPU AHB bus.		

5 SPECIFIC REQUIREMENTS OF FUNCTIONNALITY

5.1 AHB AMBA 2.0 REQUIREMENTS

5000	Wait States insertion shall be activated by the HREADY signal on slave side, whe HREADY signal is LOW.		

5010	The request on the master side shall be driven by the access type on the slave sid If the access type is INCRx (length defined) the request shall be asserted unt master is granted on the bus.		

5020	The request on the master side shall be driven by the access type on the slave sid If the access type is INCR (length not defined) the request shall be asserted unt the end of transfer.		

5030	On the master side the first transfer shall be performed when HGRANT = '1' an HREADY = '1', a transfer must be performed on the slave side when HSEL = ' and HREADY = '1' (input signal on the slave side).		

5.2 AHB SLAVE INTERFACE

5040	ERROR response shall be sent on slave side if HBURST signal on slave is differer of SINGLE (="000"), INCR (="001") and INCR4 (="011").		

5050	ERROR response shall be sent on slave side if HBURST and HSIZE does not hav the right combination (See 5040, 5100, and 5110)		

5055	Deleted		

5060	In the AHB slave I/F shall support only ERROR and OKAY responses.		

5070	In case of a valid posted write, the AHB slave I/F shall always answer OKA response. Other responses (SPLIT, RETRY and ERROR) are not used by the AH slave I/F.		

5075	If an error response is replied to a posted write, the AHBR shall send a interruption and the INTPReg field corresponding to the master having sent th transfer is set.		

5076	AHBR shall include 3 registers for interrupt management : INTPReg : interrupt pending register which stores the HMASTER number in cas of a write posted answer with HRESP=ERROR by master side. INTPReg is also		

	also a force register : writing 1 in a bit of this register triggers the amba_e interrupt. INTClearReg : Interrupt clear register. Writing '1' in the HMASTER bit, clears th bit. INTMaskReg : Interrupt Mask register. When '0', the corresponding INTPReg fiel doesn't throw interrupt.		

5080	Posted write shall be used only for all accesses : SINGLE or BURST on the slav side.		

5090	ERROR response shall be sent on slave side if IOINHIB is activated **and** HADD is within the address range [plow;phigh]. plow/phigh are 32-bit constants, however comparison is only on the word address (31:2).		

5091	ERROR response shall be sent on slave side if incremental burst of undefine length exceeds 4 words for both write and read access.		

5092	Errors from 5090 and 5091 requirements shall be store in the ERRSTATReg (Erro status register). Errors are cleared by writing '1' in the ERRSTATClearReg		

5093	AHBR registers shall be accessible through an APB slave interface		

5.3 AHB MASTER INTERFACE

5100	In SINGLE access, all types of transfer shall be accepted : byte, half word an word.		

5110	In BURST mode only word size shall be supported.		

5120	Only burst length lower or equal than 4 words shall be supported		

5130	HTRANS = BUSY (= "01") shall not be used by the master AHB I/F.		

5140	The AHB master I/F shall support only ERROR and OKAY response.		

5.4 INTERCONNECTION BLOCK

5150	If an ERROR is received on master side the ERROR response shall be given fo the same access on the slave side in READ mode.		

5160	The SINGLE or BURST mode shall be defined by the slave side: on the maste side the same access shall be performed.		

5.5 DESIGN REQUIREMENTS

5170	Internal registers can be used instead of FIFO with internal memory block, t simplify design.		

5180	The transfers from CPU to IO bus side and vice versa require resynchronisatio They shall be optimised to minimise latency.		

5190	The master clock shall have a defined phase with the slave clock as depicted i Figure 2. Only two cases are possible : • CKIO has the same frequency as CKCPU • CKIO is obtained by dividing CKCPU by 2		

5192	CKIO rising edge is roughly synchronous with CKCPU falling edge The jitte depicted in Figure 2 shall be smaller than +1/-1ns.		

5194	CKCPU shall have a ratio better than 40/60.		

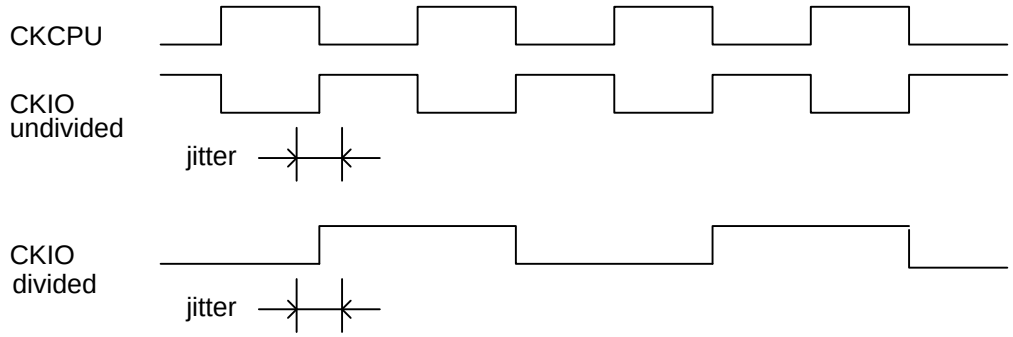

Figure 2 – Clock frequency delay between AHB CPU and AHB IO bus

5.6 CONFIGURATION REGISTERS

5.6.1 Registers List

APB address offset	Register name
0x00	INTPREG
0x04	INTCREG
0x08	INTMASKREG
0x0C	ERRSTATREG
0x10	ERRSTATCLEARREG

5.6.2 Registers Detailed Definition

Bits	Field Name	Description	R/W	Reset value
31..16	RESERVED	Reserved	R	0
15..0	AMBA_ERR	Indicate that an HRESP_ERROR as been replied to a posted write performed by the corresponding CPU master number. bit 0 = "1" => HRESP_ERROR replied to a posted write access by CPU master n°0 bit x = "1" => HRESP_ERROR replied to a posted write access by CPU master n°x INTPReg is also a force register. Writing 1 in a bit of this register triggers th amba_err interrupt signal.	R/W	0

Table 1 : [0x00] AHBR Interrupt Pending and Force Register [INTPREG]

Bits	Field Name	Description	R/W	Reset value
31..16	RESERVED	Reserved	-	0
15..0	CLEAR_ERR	Writing a "1" clears the corresponding INTPREG field.	W	0

Table 2 : [0x04] AHBR Interrupt Clear Register [INTCREG]

Bits	Field Name	Description	R/W	Reset value
31..16	RESERVED	Reserved	R	0
15..0	MASK_ERR	Writing a "1" unmask the corresponding INTPREG field.	R/W	0

Table 3 : [0x08] AHBR Interrupt Mask Register [INTMREG]

Bits	Field Name	Description	R/W	Reset value
31..2	RESERVED	Reserved	R	0
16..1	LONGBURST_ERR	An HRESP_ERROR has been replied because a burst access longer than 4 accesses has been requested through the AHBR.	R/W	0
15..0	IOINHIB_ERR	An HRESP_ERROR has been replied because of an access conflicting with IOINHIB value.	R/W	0

Table 4 : [0x0C] AHBR Error Status Register [ERRSTATREG]

Bits	Field Name	Description	R/W	Reset value
31..2	RESERVED	Reserved	-	0
1..0	CLEAR_ERR	Writing a "1" clears the corresponding ERRSTATREG field.	W	0

Table 5 : [0x10] AHBR Error Status Clear Register [ERRSTATCLEARREG]

5.7 INTERFACES

Signal	*Dir.*	*Src/Dest*	*function*	*Reset value*
Clocks & resets				
CKCPU	In	Slave	Slave interface clock	-
CKIO	In	Master	Master interface clock	-
RstCPU_N	In	Slave	Reset of Slave interface	-
RstIO_N	In	Master	Reset of Master interface	-
PLOW[31:0] PHIGH[31:0]	In	Slave	Defines the range for the IONHIB access protection.	-
IOINHIB	In	Slave	When set, inhibition of accesses outside of th memory range [plow;phigh]	-
CPUIORATIO[1 :0]	In	Slave	Indicates frequency ratio between CPU AHB Bus an IO AHB Bus	-
Slave interface input signals (ahbsi)				
HSEL	In	Slave	Slave selection signal active to '1'	-
HWDATA[31:0]	In	Slave	Write data on slave I/F	-
HWRITE	In	Slave	Indicates a write transfer '0' : read access '1' : write access	-
HSIZE[2:0]	In	Slave	Defines data transfer size	-
HPROT[3:0]	In	Slave	Indicates protection control	-
HTRANS[1:0]	In	Slave	Defines transfer types "00" : IDLE "01" : BUSY "10" : NONSEQ "11" : SEQ	-
HBURST[2:0]	In	Slave	Indicates burst type transfer	-
HADDR[31:0]	In	Slave	Address Slave bus	-
HREADY	In	Slave	HREADY slave input	-
HMASTER[3:0]	In	Slave	Indicates which bus master is currently performing	-
HMASTLOCK	In	Slave	Indicates that the current master is performing locked sequence of transfers	-
Slave interface output signals (ahbso)				
HSPLIT[15:0]	Out	Slave	Indicates to the arbiter which bus masters should b allowed to re-attempt a split transaction	00…
HREADYOUT	Out	Slave	Indicates that a transfer has finished on the bus	1
HRESP[1:0]	Out	Slave	Provides transfer response "00" : OKAY "01" : ERROR "10" : RETRY "11" : SPLIT	00

HRDATA[31:0]	Out	Slave	Read data on slave I/F	00…
Master interface input signals (ahbmi)				
HGRANT	In	Master	Indicates that the master is currently the highes priority master	-
HREADY	In	Master	Indicates that a transfer has finished on the bus	-
HRESP[1:0]	In	Master	Provides transfer response "00" : OKAY "01" : ERROR "10" : RETRY "11" : SPLIT	-
HRDATA[31:0]	In	Master	Read data Master bus	-
Master interface output signals (ahbmo)				
HBUSREQ	Out	Master	Indicates that the master requires the bus	0
HLOCK	Out	Master	Indicates that the master requires locked access to th bus	0
HWRITE	Out	Master	Indicates a write transfer '0' : read access '1' : write access	0
HTRANS[1:0]	Out	Master	Defines transfer types "00" : IDLE "01" : BUSY (not supported) "10" : NONSEQ "11" : SEQ	00
HBURST[2:0]	Out	Master	Indicates burst type transfer	000
HSIZE[2:0]	Out	Master	Defines data transfer size	010
HPROT[3:0]	Out	Master	Indicates protection control	0000
HADDR[31:0]	Out	Master	Address Slave bus	00…
HWDATA[31:0]	Out	Master	Write data Master bus	00…
Interrupt signal				
amba_err	Out	Interrupt Controller	Interrupt AHB error signal	0

6 ARCHITECTURE DESCRIPTION

6.1 SLAVE LAYER DESCRIPTION

The slave layer receives all AHB signals of the CPU bus. This module manages AHB signals necessary in order to respond at master request.

Otherwise, all control signals between slave layer and interconnection block are generated or received in this module.

Combinatory signals shall be used to generate and control signals interface. A finite state machine manages HREADY output signal and other internal signals (Figure 3).

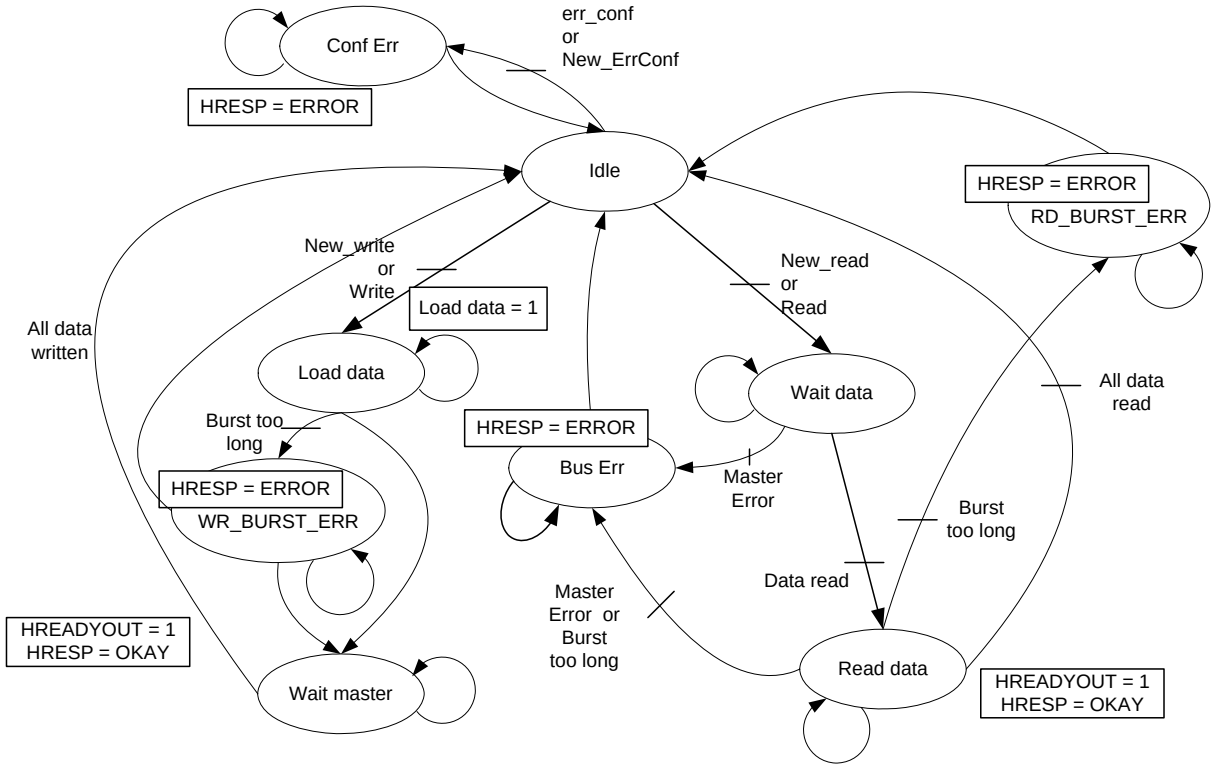

Figure 3 – Slave finite state machine

Idle: Waiting access state

- When HTRANS = NS, HREADY = '1', HSEL = '1' and HWRITE = '1' (write access), the next state is Wait master.
- When HTRANS = NS, HREADY = '1', HSEL = '1' and HWRITE = '0 (read access)', the next state is Wait data.

- When a mismatch configuration, HSIZE > 32 bit, HBURST > INCR4, is detected the next state is Conf Err.

Conf Err: this state generates error response on the CPU bus when a mismatch configuration is detected, thus response is active during two cycles clock. The next state is Idle.

Load data: load data signal is activated then AHB data is buffered in internal register. The next state is Wait master.

BURST_ERR: A 2 cycle error response is performed when a write burst incremental access exceeds 4 words.

Wait master: This state is a waiting state, when data is written on master interface. The next state is idle. If an error is detected on master side the next state is Write err.

Bus Err: this state generates error response on the CPU bus when an error response is received by the master side when an Read access is performed or when a burst incremental access exceeds 4 words. This response is active during two cycles clock. In case of a Write access, error response is not sent (due to posted write access mechanism) but an interrupt is generated and the master number which initiated the access is memorized in the INTPReg register. The next state is Idle.

Wait data: it is a waiting state, when a data is available the next state is Read data.

Read data: HREADYOUT signal is asserted and HRESP = OKAY. If no new access and all data are written the next state is Idle otherwise the next state is Load data.

6.2 MASTER LAYER DESCRIPTION

The master layer receives all AHB signals of the IO bus. This module manages AHB signals necessary in order to manage slave interface on the IO bus request and the first stage of signals for interface data and control for the SLAVE/MASTER I/F.

For this module, combinatory signals and finite state machine are used to manage master layer.

Following finite state machine synopsis manage output and internal design signals (Figure 4):

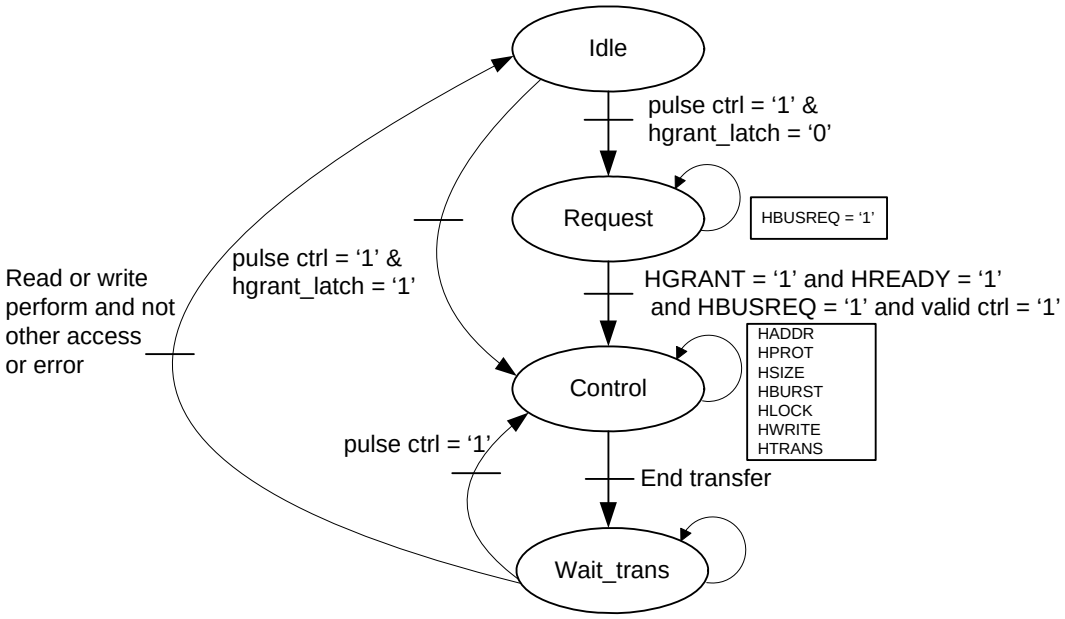

Figure 4 – Master finite state machine

Idle: Waiting state, when a new access is performed the next access is Request.

Request: In this state, the HBUSREQ signal is asserted, when HGRANT, HREADY HBUSREQ and valid ctrl is active, the next state is Control.

Control: This state manages all AHB control signals.

- HTRANS signal with this different access type (Idle, Non Seq, Seq).
- In Burst mode, the address increment

When end transfer, the next state is wait.

Wait_trans: Waiting state, waiting a new transfer

- If a new transfer is required the next state is Control
- If no other transfer or error response the next state is Idle

6.3 SLAVE/MASTER INTERFACE

The SLAVE/MASTER interface allows exchange between Slave layer and master layer. In this module, data and controls are buffered and synchronised on different clocks domains according to direction of controls and data. Two different data path are used one for the write data, data from slave to master, and one for the read data, data from master to slave.

This module generates all internal signals necessary to generate pulses. These pulses indicate when data and control are valid. This module generates all AHB output signals.

To manage all signals shall be used synchronous or asynchronous process.

Different blocks are used in write and read data path

- Pulse generator
- Edge detection
- Pulse detection

Pulse generator:

This module generates a pulse with a length adapted to master side frequency clock. With this mechanism the pulse is detected by the master side.

If the ratio between CPU AHB bus and IO AHB bus is two, pulse generates on the slave side frequency must be two clock period.

Edge detection:

This module allows edge pulse detection and generates a pulse used in finite state machine.

Two registers is used for all AHB control signals to manage a chain access on the bus (two non Sequential access), the write is posted so the second access is ready and the AHB Control signals changes.

So it is necessary to use a another register to save the current AHB control signals and load the newt AHB control signals

6.3.1 SYNOPSIS

The following synopsis shows the bridge architecture (Figure 5):

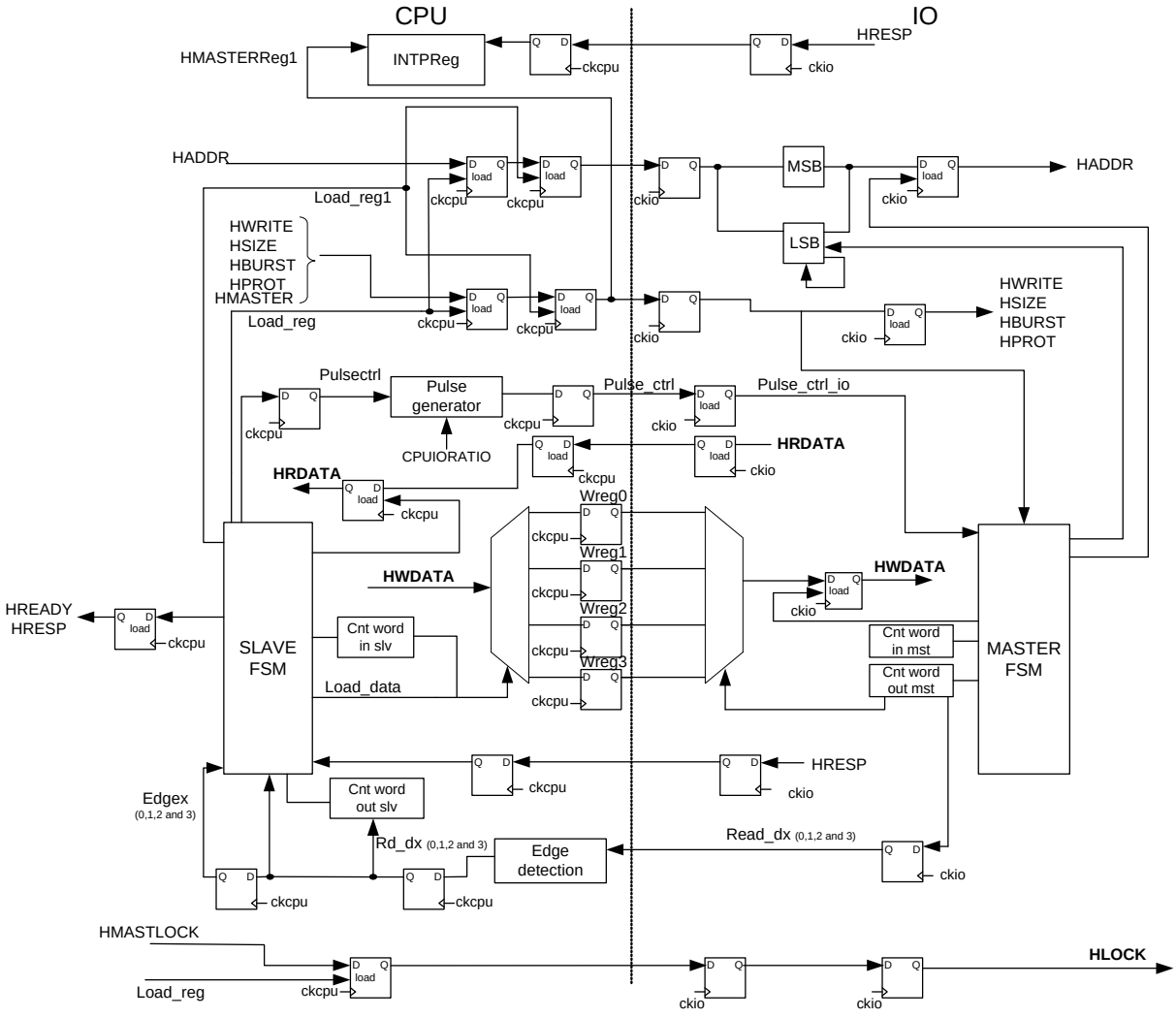

Figure 5 – AHB to AHB bridge architecture

6.3.2 WRITE DATA PATH

This architecture uses the handshake behaviour, data and control valid signal are generate on the frequency domain and use by other frequency domain.

Four registers are used to buffer data. 4 words burst of 32 bits must be performed with these four registers in burst mode. An interruption signal must be generated when an error response is received.

Slave FSM manages all output signals necessary to data buffer, AHB control signals buffer (HWRITE, HPROT, HMASTLOCK, HSIZE, HBURST and HADDR).

With the Slave FSM, when HWRITE, HPROT, HMASTLOCK, HSIZE, HBURST and HADDR are valid a pulse ctrl signal is sent in master side, and then the data is buffered in registers. Master FSM is awaked by this signal. When data are valid (Valid_d signal is active) this data is performed in master side.

When data are performed on master side, FSM master send Pulse read x signal to indicate on slave side that data are performed. Then the master side is ready for other transfer.

See the following single access timing with the same frequency on each bus (Figure 6), single access timing with CPUIORATIO = "01" 2x (Figure 7), burst access timing(Figure 8).

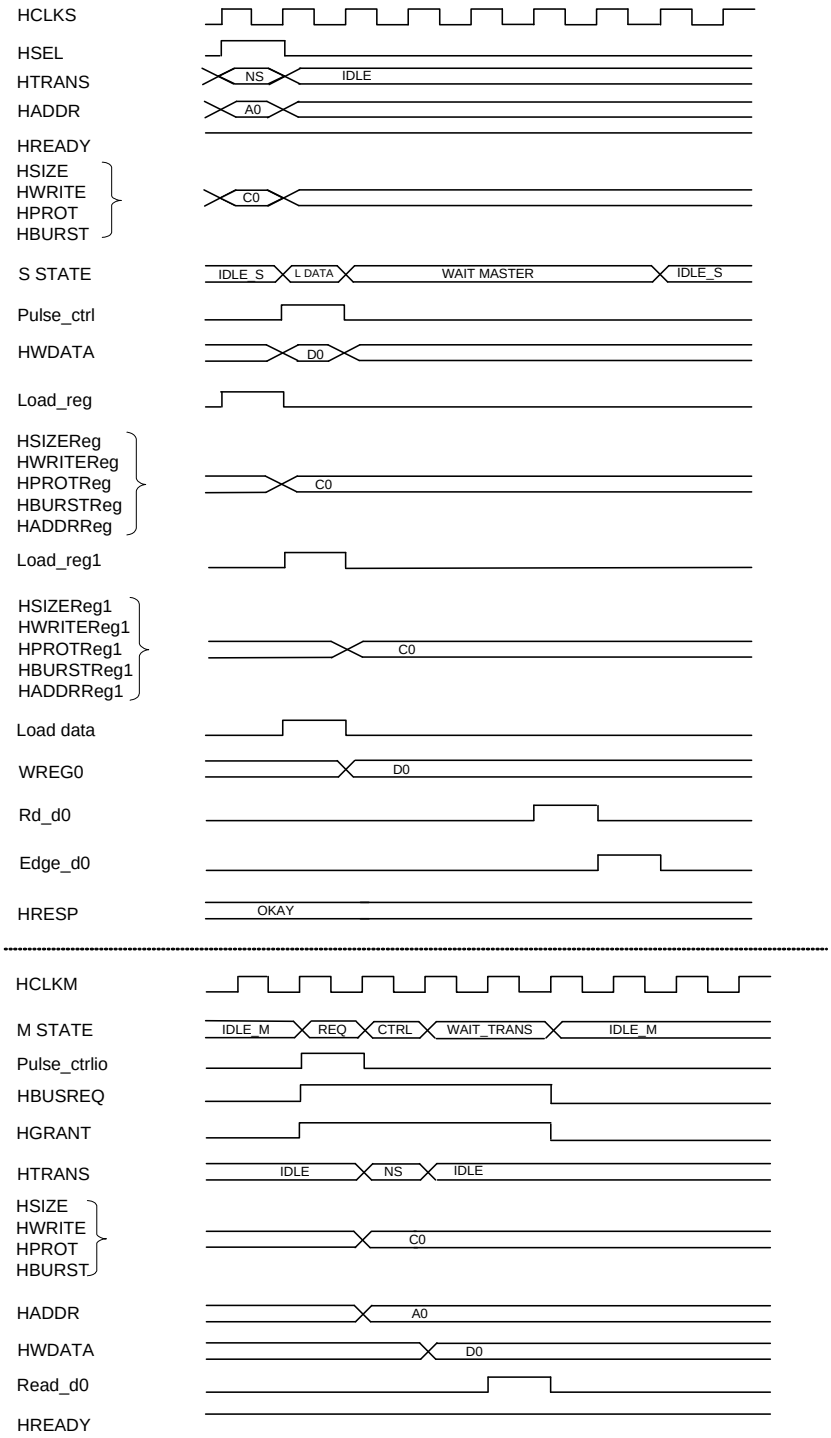

Figure 6 – Write single access timing (CPUIORATIO = "00", 1x)

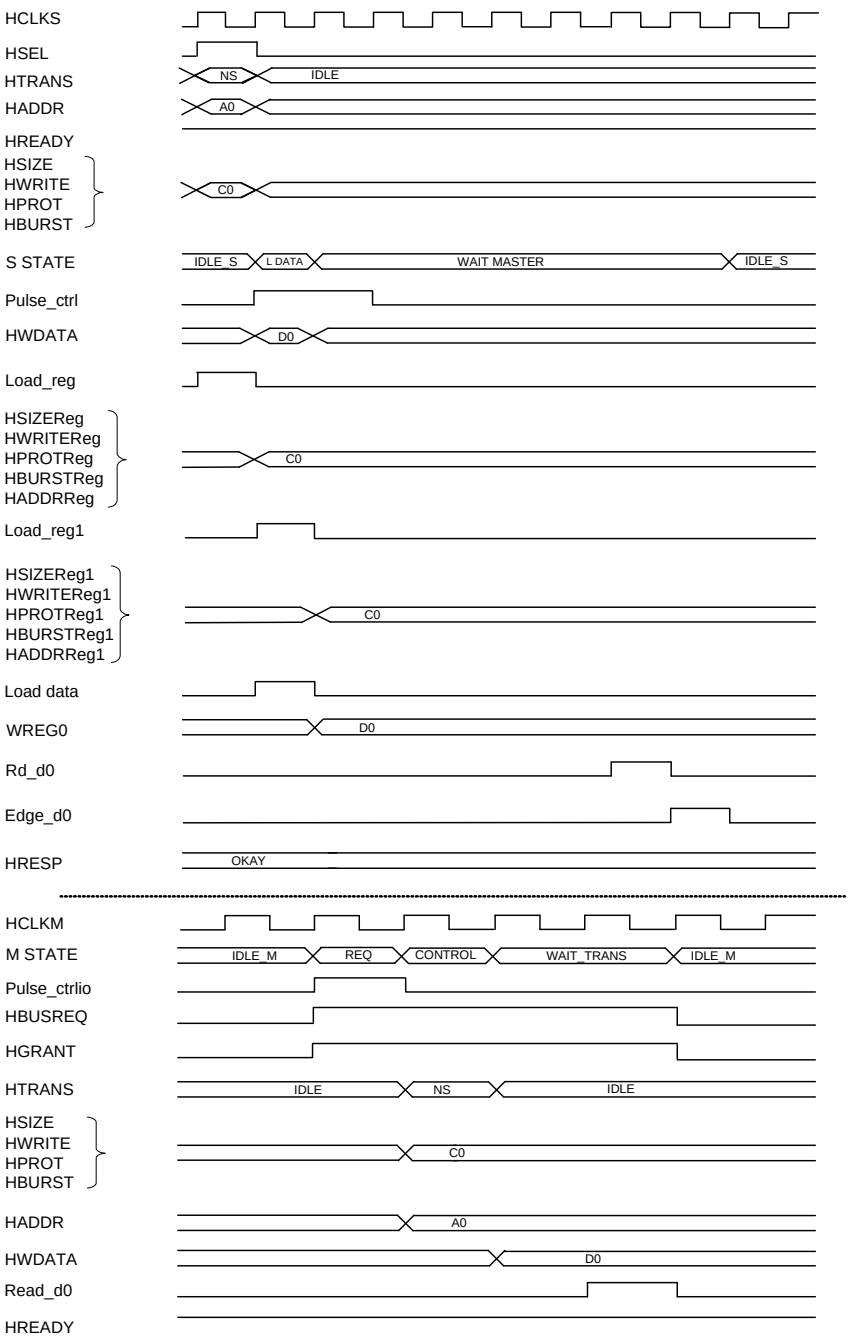

Figure 7 – Write single access timing (CPUIORATIO = "01", 2x)

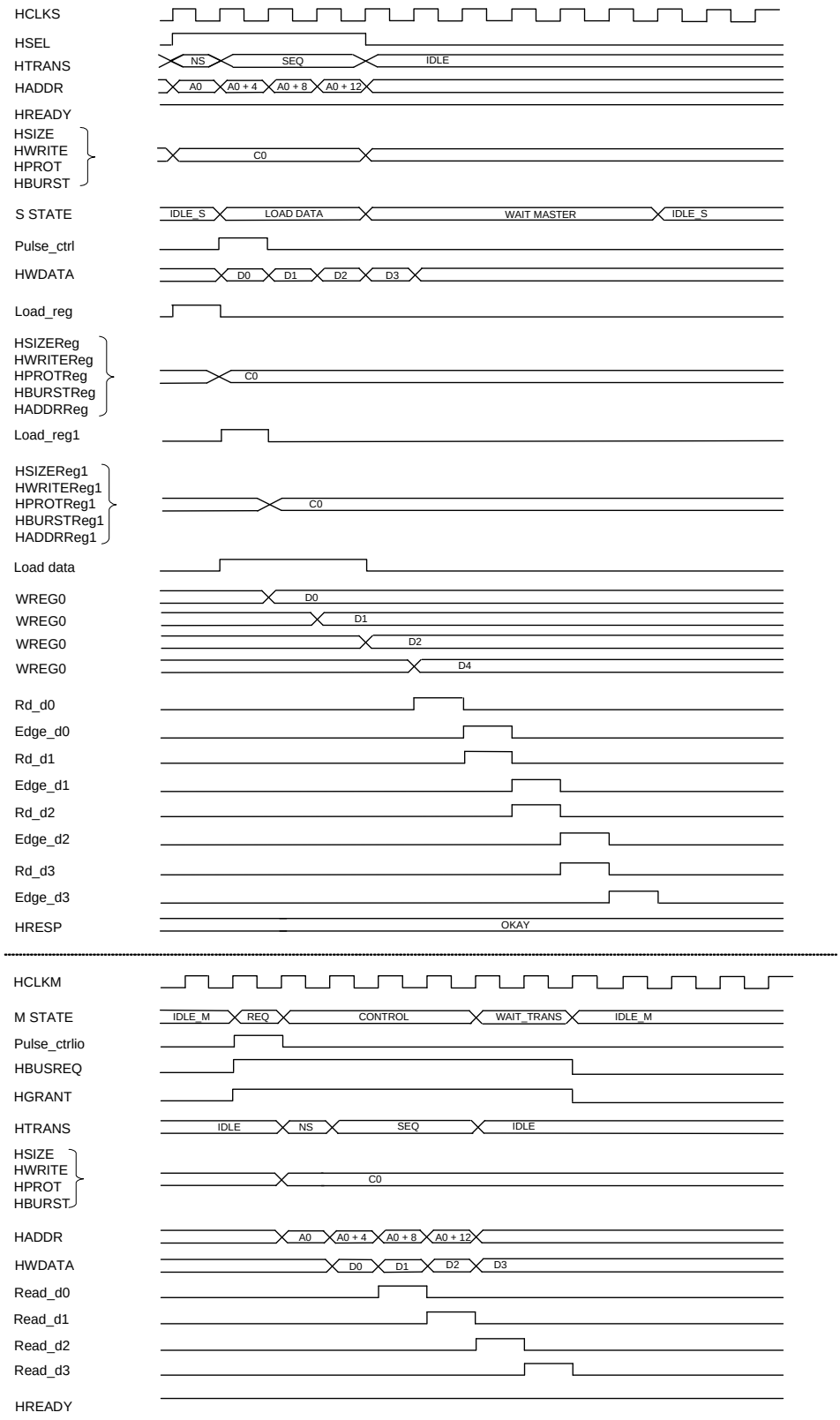

Figure 8 – Write burst access timing (CPUIORATIO = "00", 1x)

6.3.3 READ DATA PATH

This architecture uses the handshake behaviour, data and control valid signal are generate on the frequency domain and use by other frequency domain.

Slave FSM manages all output signals necessary to buffer data, buffer AHB control signals (HWRITE, HPROT, HMASTLOCK, HSIZE, HBURST and HADDR).

With the Slave FSM, when HWRITE, HPROT, HMASTLOCK, HSIZE, HBURST and HADDR are valid a pulse ctrl is sent in master side and the data is buffered in register. Master FSM is awaked by this signal.

When a single access is performed on the slave side a single access is performed on the master. If a burst access is performed in slave side, a 4 word burst access performed in master side.

When the first data is valid a signal is send in slave side to performed the data transfer.

If an ERROR is received on HRESP Master side signal, this error must be transmitted in slave interface to indicate this error on the CPU AHB bus.

See the following single access timing with the same frequency on each bus (Figure 9), single access timing with CPUIORATIO = “01” 2x (Figure 10) and burst access timing (Figure 11) .

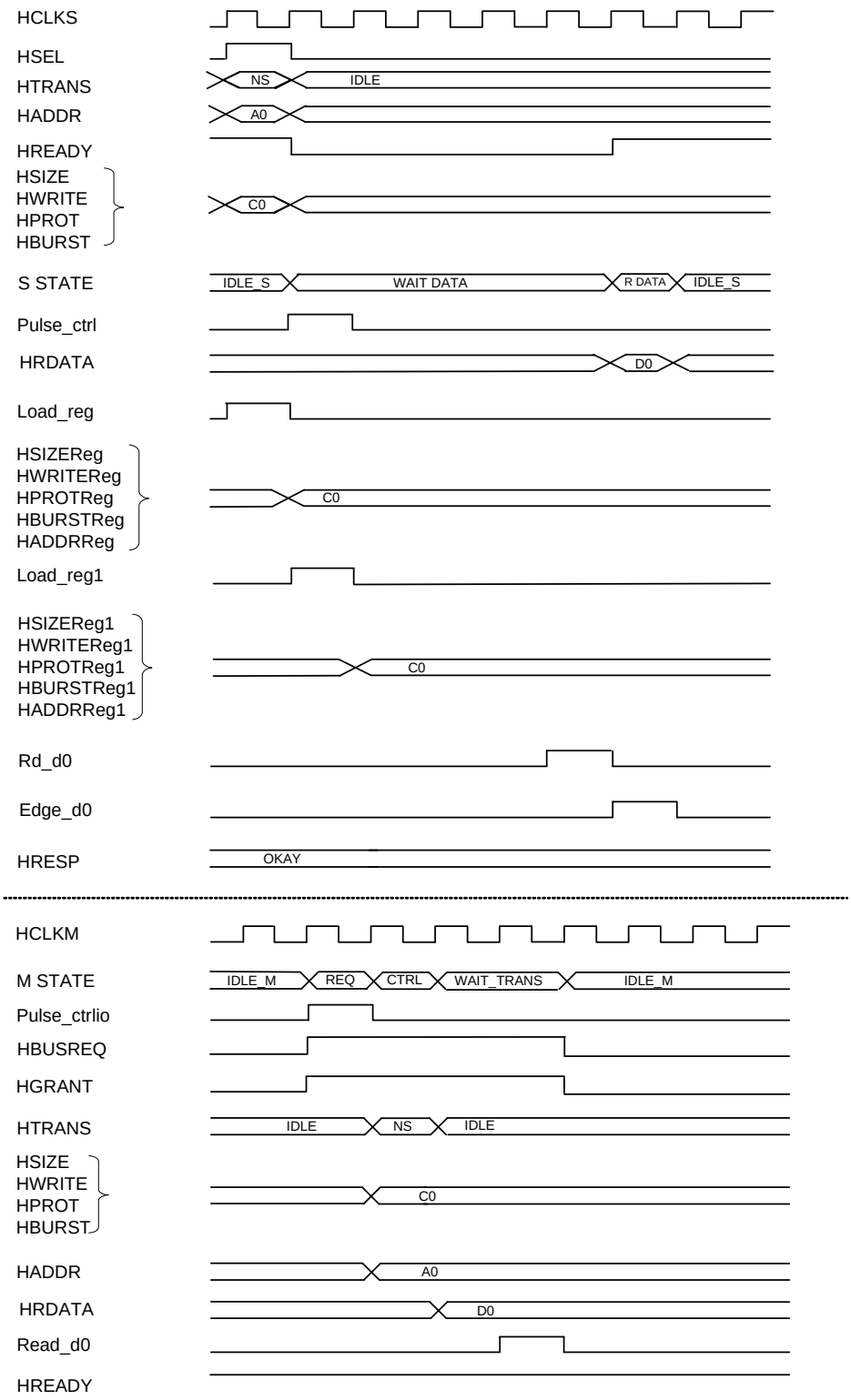

Figure 9 – Read single access timing (CPUIORATIO = "00", 1x)

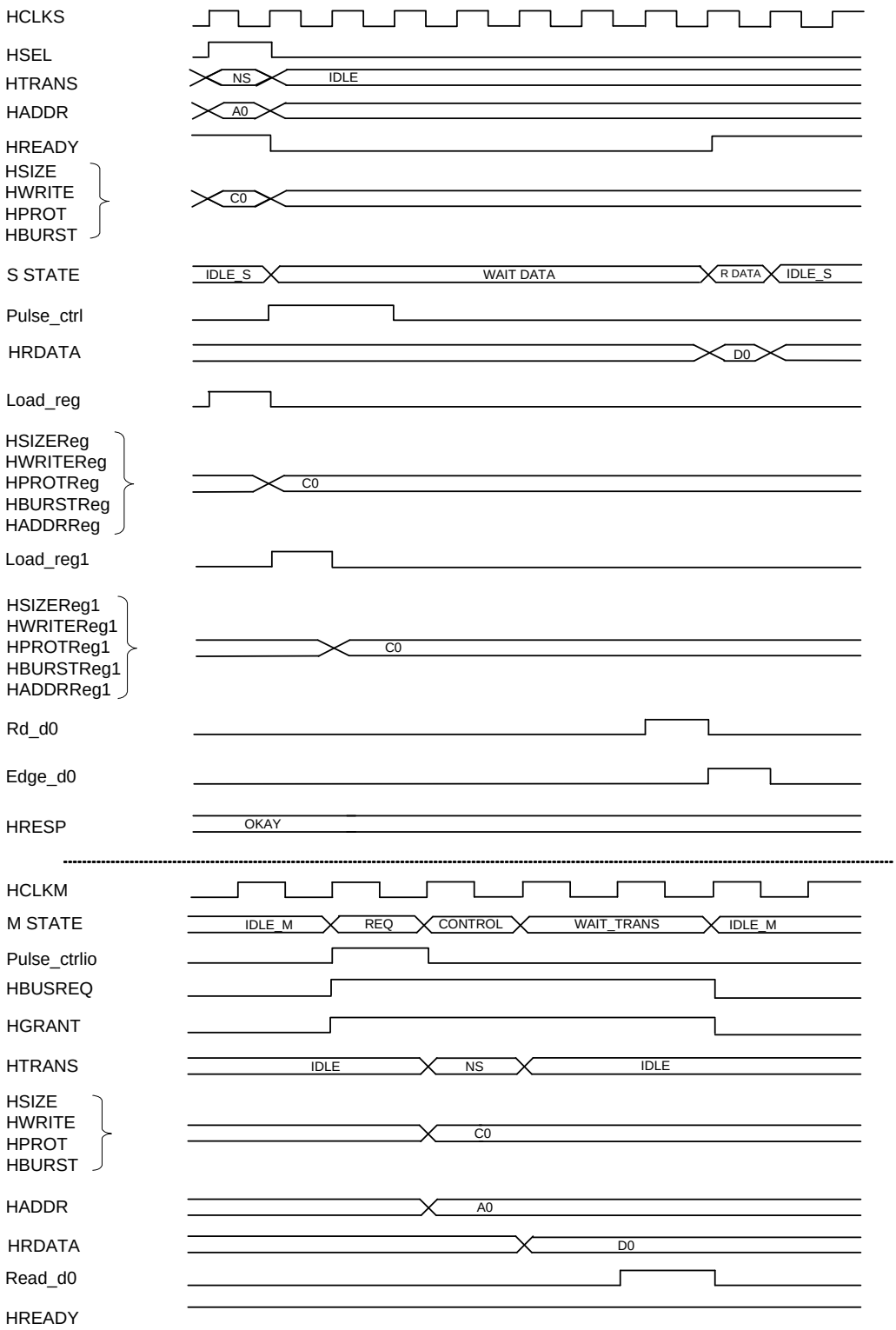

Figure 10 – Read single access timing (CPUIORATIO = "01", 2x)

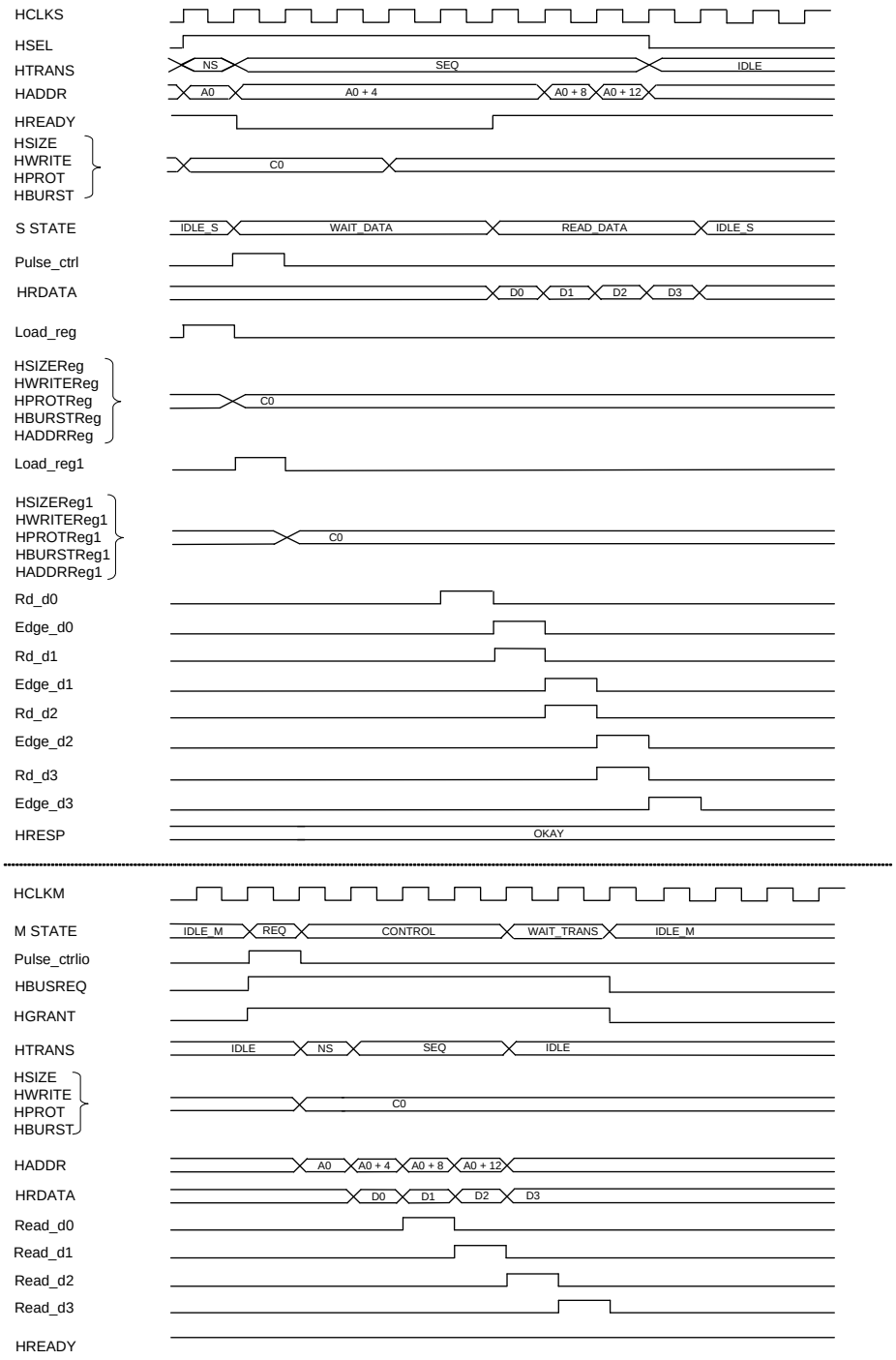

Figure 11 – READ burst access timing (CPUIORATIO = "00", 1x)

7 SYNTHESIS RESULTS

7.1 SYNOPSYS SYNTHESIS RESULT

Use Technology of ATMEL ATC18RHA

Total cell area = 68013 µm²

A Nand gate is 12.54 µm²

Count gate = 68013 / 12.54 = 5424 gates (NAND2-equivalent) of which 600 flip-flops

Frequency ~ 130 MHz (CKCPU) and 145 MHz (CKIO) pre-layout with wire load model 8KG

7.2 RTAX SYNTHESIS RESULT

Use technology RTAX400s_cqfp352-s

Combinational Cells: 1906 (5%), Sequential Cells: 684 (3%), Total Cells: 2590 (5%), Block Rams: 0

The performance result are 63.6 MHz for CKCPU and 62.2 MHz for CKIO.

DISTRIBUTION LIST

	Overall document		Summary
	Action	Information	