Synthesis of ATM Switch Controller Modules with the Protocol Compiler$^{\text{TM}}$ from Synopsys$^{\text{TM}}$

Walter Lange Wolfgang Rosenstiel
Lehrstuhl für Technische Informatik, Universität Tübingen
Sand 13, D 72076 Tübingen, Germany
Telephone number: 0049-7071-2975482
E-mail: <wlange|rosen>@informatik.uni-tuebingen.de

Abstract

In order to manage the higher complexity of VLSI chips and to reach shorter design cycles, the design effort becomes increasingly focused on higher levels of abstraction. We describe the modeling of some modules of an high speed telecommunication circuit, an ATM Switch Controller (ASC) using the Protocol Compiler or Dali$^{\text{TM}}$ *from Synopsys*$^{\text{TM}}$*. Dali*$^{\text{TM}}$ *supports a fast and compact graphical entry of a protocol controller hardware description with graphical signs similarly to formula symbols. The output of the synthesis with Dali*$^{\text{TM}}$ *is simulated. Advantages of this design method are discussed and the results of the synthesis are presented.*

Contents

1 Introduction 4

2 Design of the ATM Switch Controller (ASC) 4

3 Design Method with Dali 5

3.1 General 5
3.2 Dali Frames 6
3.2.1 The Used Template Frames: 7
3.3 Dali Actions 8
3.3.1 User Defined Actions 8
3.3.2 Dali Version and Modeling Steps 8
3.4 Options Settings for the ATM Design 9
3.4.1 Controller Styles 9
3.4.2 Single or Partitioned Controller 9
3.4.3 Min Encoding: Map Unreachable States 10
3.4.4 Loop Handling, “Use Counter” Option 10
3.4.5 Effects of Option Setting 10

4 Modeling five ASC-Modules with Dali 10

4.1 Modeling of the ASC Input Module AHT_IN 12
4.1.1 The AHT_IN Frame Hierarchy 13
4.1.2 The AHTIN-FSM’s 14
4.1.3 Simulation of AHT_IN 14
4.1.4 Summary of Synthesis Results of Module AHT_IN 14
4.2 Modeling of the Header Translator Module HT 16
4.2.1 The HT frames 16
4.2.2 The HT FSM 17
4.2.3 Simulation of the Header Translator 17
4.2.4 Synthesis Results of Module HT 18
4.3 The Shift Register Control Module (SRC) 19
4.3.1 The SRCIN Frame 19
4.3.2 The “load_reg_encode” and the “load_shreg” Frames 20
4.3.3 The “shift_payload” Frame 20
4.3.4 The SRC-FSM 21
4.3.5 Simulation of the Module SRC 22
4.3.6 Synthesis Results of Module SRC 22
4.4 The Receive Data Module (RD) 23
4.4.1 The RDIN Frame 24
4.4.2 The “receive_cell” Frame 24
4.4.3 The “store_cell” Frame 24
4.4.4 The “decode_reg” Frame 25
4.4.5 The “store_reg” Frame 25
4.4.6 The FSM Graphs 25

4.4.7 Simulation of the Module RD 26
4.4.8 Synthesis Results of Module RD 26
4.5 The AHT Output Module (AHT_OUT) 28
4.5.1 The AHTOUT Frame 28
4.5.2 The “fetch_cell” Frame 29
4.5.3 The “xmit_cell” Frame 29
4.5.4 The “xmit_4bytes” Frame 29
4.5.5 The FSM Graphs 30
4.5.6 Simulation of the Module AHT_Out 30
4.5.7 Synthesis Results of Module AHT_Out 31

5 Summary of the Results 32

6 Conclusion 33

7 Acknowledgment 33

8 Attachment: PC Synthesis Reports 34
8.1 Input Module AHT_IN 34
8.1.1 The Dali Synthesis Report for AHT_IN 34
8.1.2 The Logic Synthesis Report for AHT_IN 36
8.2 Header Translator Module HT 40
8.2.1 The Dali Synthesis Report for HT 40
8.2.2 The Logic Synthesis Report for HT 41
8.3 Shift Register Control Module SRC 45
8.3.1 The Dali Synthesis Report for SRC 45
8.3.2 The Logic Synthesis Report for SRC 46
8.4 Receive Data Module RD 50
8.4.1 The Dali Synthesis Report for RD 50
8.4.2 The Logic Synthesis Report for RD 51
8.5 The AHT Output Module AHT_Out 56
8.5.1 The Dali Synthesis Report for AHT_Out 56
8.5.2 The Logic Synthesis Report for AHT_Out 58

9 References 63

1 Introduction

The rising complexity and structure density as well as shorter development cycles in VLSI chip design require Design Automation (DA) on higher levels of abstraction [Gaetal96].

The Protocol Compiler Dali^{TM} from Synopsys^{TM} allows the modeling of protocol control logic in ASIC designs using a graphical design method. Dali generates Finite State Machines with integrated data handling actions. The output of the Protocol Compiler is a simulatable RTL-VHDL description which can be further synthesized by an appropriate tool like the Design Compiler^{TM} from Synopsys^{TM}.

The application used in this report is an ATM Layer circuit.

ATM (Asynchronous Transfer Mode) is considered the ultimate data transmission method by the telecommunication industry for Broadband ISDN [Pryk93], it is still in the process of standardization.

ATM is parted into three layers (similarly to the three lower layers of OSI): *the Physical Layer, the ATM Layer and the ATM Adaption Layer*. The ATM data transmission is performed in data packets, so called “cells” with a constant length of 53 bytes. The cell header has a length of 5 Bytes and the cell payload has a length of 48 bytes. In a telecommunication node, ATM cells are accepted serially in the Physical Layer where the cell header is recognized. The cell is parallelized and handed over to the ATM Layer where the cell is *routed* through an ATM switch according to some information in the cell header. The electronic circuit performing the ATM Layer functions is called the ATM Switch Controller (ASC).

2 Design of the ATM Switch Controller (ASC)

Figure 1 shows roughly the block circuit of the ATM Layer for 14 channels.

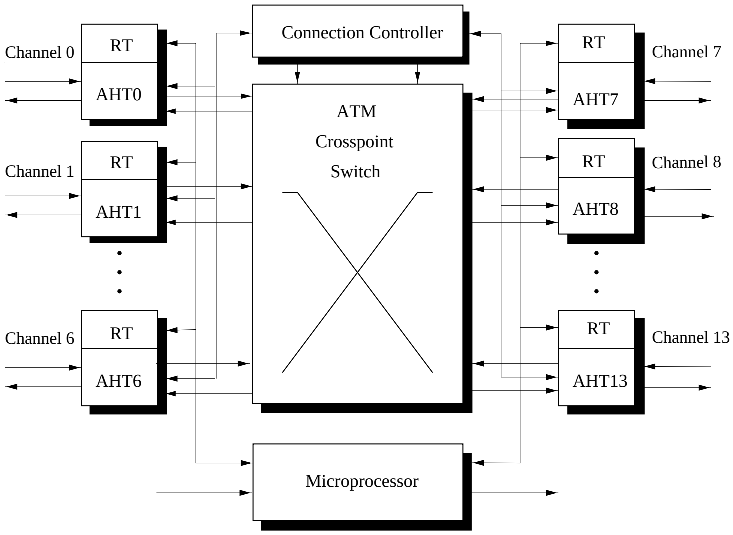


Figure 1: ATM Layer Circuit

The parallelized ATM cells arrive from the Physical Layer in the Address Header Translators AHT0 to AHT14. The cell header of each cell in each AHT is separated and translated according to the information in the Routing Table (RT) to a new cell header with the routing information attached. The cell with the new header plus routing information (which indicates the target output channel) is stored in a cell buffer and, if the output channel is free, routed through the ATM Switch to the proper output channel. The Connection Controller controls the ATM Switch, it receives the routing requests from the AHT's, maintains tables of the busy and free connections and configures the switch if a new connection is due. The microprocessor has administrative tasks, it performs the reset of all circuits after power on, receives the signaling cells which require a dynamic Routing Table update in order to establish new point to point connections. The planned ATM switch is a commercially available one-chip "non-blocking ATM Crosspoint Switch" (TriQuint TQ8017) with a data rate of 1.2 GBit/s per channel.

The design of the ASC is initially performed with structural methods using the Hardware Description Language VHDL [LaRo97]. The complete design is synthesized with the Behavioral Compiler^{TM} [LaRo99-1] [LaRo99-2]. Five modules are described and synthesized again with Dali^{TM}.

Figure 2 shows the block diagram of one ASC channel plus Connection Controller. Five modules (with shade) are described and synthesized again with Dali^{TM}.

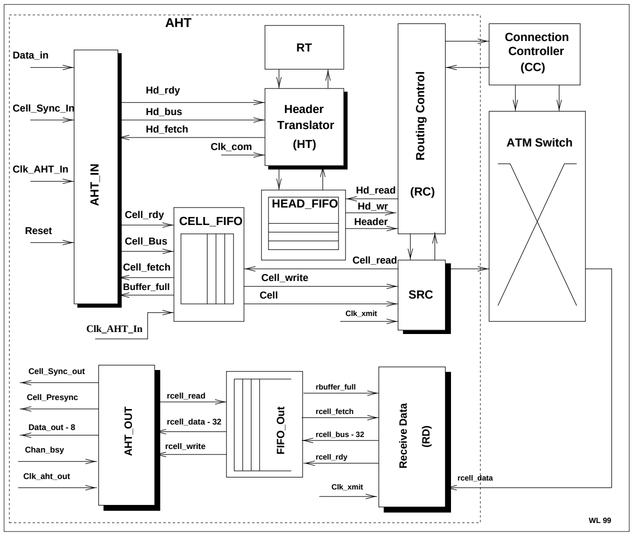


Figure 2: Block Diagram of one ASC channel

3 Design Method with Dali

3.1 General

Dali offers a GUI for the modeling of hardware modules: Refer to Protocol Compiler User Guide [SynPcUG98], Controller Design Using Protocol Compiler [SynCDPC98] and Dali Tutorial for VHDL users [DaTut97].

The Dali GUI supports a fast and compact graphical entry of a protocol controller hardware description offering a set of modeling symbols called frames. The advantage of this method is, that the functions and interfaces which are represented graphically, can be described, comprehended and checked faster than a VHDL text.

The disadvantage is, that a new non standardized modeling method has to be learned and trained.

3.2 Dali Frames

Dali provides the following basic frames (refer to figure 3):

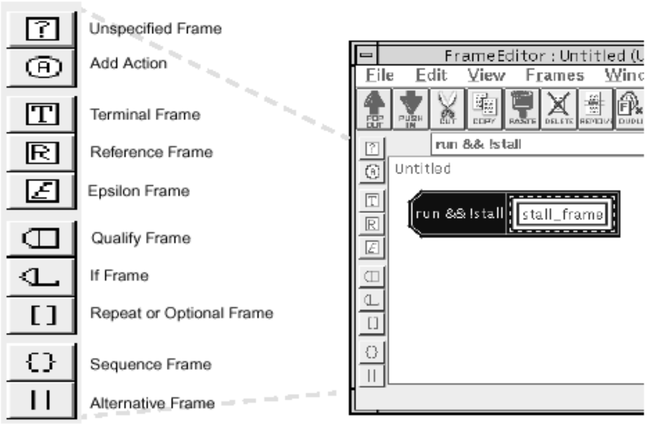


Figure 3: Types of Dali frames and Frame operators

Hierarchical Frame Types:

1. Unspecified Frame
2. Terminal Frame
3. Reference Frame
4. Template Frame
5. Epsilon Frame

Hierarchical Frame Operator Types:

1. Qualify Frame Operator
2. IF Frame Operator

3. Repeat or Optional Frame Operator
4. Sequence Frame Operator
5. Alternative Frame Operator
6. Simultaneous Frame Operator
7. RunIdle Frame Operator
8. Exception Frame Operator

3.2.1 The Used Template Frames:

Template frames are frame definitions with parameters. This frame type can be used in multiple places within a protocol customized by its parameters.

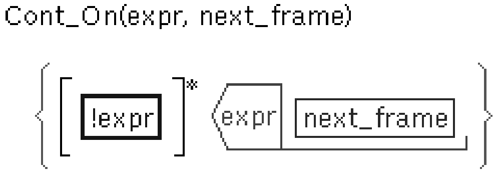


Figure 4: The Cont_on() Template

In our design the following templates are used:

- Cont_On(expr, next_frame): Has a “Wait” function for the signal named “expr” and executes the “next_frame”- frame thereafter (refer to figure 4).
- Proc_On(start_expr,task_to_execute): Defines an endless process (see figure 5): Waits until start_expr is high (level sensitive). The task_to_execute is executed endlessly until it terminates.

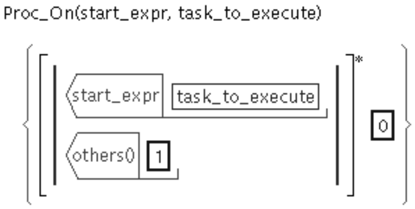


Figure 5: The Proc_on() Template

3.3 Dali Actions

Dali actions are defined as simple data path operations, which can be attached to a frame and executed when the frame is accepted during design execution. Dali provides a set of “built-in actions” like

- set(x): sets the value of x to ’1’
- clear(x): sets the value of x to ’0’
- Assignment, (concatenation, splitting): $x =< expr1 >, expr2 >, ...$
- Increment, decrement: incr(x), decr(x)

An action is executed in one clock cycle and can take ports, variables, expressions and constants as arguments.

3.3.1 User Defined Actions

Define customized actions to perform operations which are not available as built-in actions. User defined actions can be described as VHDL-procedures or functions in VHDL-packages. In our case we provided the encoding and decoding procedures for the modules SRC and RD as user defined actions in additional VHDL-packages.

3.3.2 Dali Version and Modeling Steps

Dali Version used: 1998.08. The modeling steps are as follows:

1. Enter the design of one module with Dali GUI. That means: enter ports, frames, variables, actions etc. for one module with Dali.
2. Generate a VHDL package for the design actions, if necessary.
3. Generate a VHDL description of the test bench for the module. Dali provides a template for the test bench. The programmer has to enter the stimuli.
4. Check the design, synthesize the design and generate VHDL code with Dali.
5. Analyze the VHDL package, the Dali-generated module code and the testbench with VSS$^{\mathrm{TM}}$-Analyzer.
6. Simulate and debug the module with the Synopsys debugger program and the wave-viewer which is called with the command “vhdldbx”.

3.4 Options Settings for the ATM Design

3.4.1 Controller Styles

The controller style specifies how the design is compiled and determines the overall encoding of the controller. Dali allows the following controller styles for a single controller:

- Distributed Encoding: Default encoding. This style produces a controller implemented as many simple, low fanin, next-state-logic functions feeding the state registers. Advantages of this encoding style are: Implements fast controllers, allows debugging, does not depend on the total number of states, creates FSM's that are similar to one-hot encoded FSM's. The disadvantages are: When used on the entire design, this style can use many registers leading to high area cost.

- Binary - Min Encoded: Minimizes the total number of states and encodes the states to use the minimum possible number of state registers. The next-state-logic functions can be very complex. Advantages: This style produces compact, area minimized designs. Disadvantages: This style gives good final gate-level results only when the total number of states is less than about 200. For the ATM design, the Binary - Min encoded style has been used for all modules.

- Min Distance - Min Encoded: The state codes are assigned to minimize the average number of changing bits per state transition. This style uses an encoding scheme similar to the Gray Code. Advantages: Creates single state graphs for an entire controller and minimizes the total number of states. Disadvantages: If used for FSM's with more than 200 states, the next state and action trigger logic becomes very complex .

- Branch and Sequencing Encoding: Similar to Binary - Min Encoding except that slightly more than the minimal number of state bits are used. States are grouped into state sequence clusters. The state registers are split into two fields: One field determining the state sequence cluster and the other field determining the state within this cluster. Advantages: Similar to Binary - Min Encoded. Disadvantages: Only for designs with less than about 200 states.

- One Hot Encoding: One state register bit represents one state of the controller. Only one state bit can be set at a time. Disadvantages: Large state vectors. When the total number of states is greater than 100, this style produces very large next-state-logic functions. The time to synthesize these function can be unacceptably long. Therefore this encoding style is not selectable and only used for the initial FSM.

The selection for the different encoding styles is in the menu edit/partition/style.

3.4.2 Single or Partitioned Controller

For some designs, especially for those described with two or more parallel endless processes, the selection “partitioned controller” delivers overall less states and smaller areas.

3.4.3 Min Encoding: Map Unreachable States

Dali adds in the "Min Encoding" option additional states and transitions so that unused state codes (unreachable states) always transition back to valid states. These unused states cannot be reached in normal operation, however if the controller enters an invalid state for some reason (e.g. due to electrical disturbances or hazards), it immediately transitions to a valid state. The "Map Unreachable States" option adds the number of unreachable states to the reachable states. In many cases, the number or states are doubled.

3.4.4 Loop Handling, "Use Counter" Option

The default selection for the synthesis are unrolled loops. Rolled loops often reduce the area. In this case a counter should be implemented (select Edit/Use Counter).

3.4.5 Effects of Option Setting

The effect on FSM states, area and delay for different option settings are shown in table 1. As an example the first module AHT_IN is synthesized. The options *Single Controller*, *Partitioned Controller*, *Use Counter*, *Map Unreachable States* are explained above. For all option combinations, the controller style "Binary - min. encoded" has been selected, since - after optimization, a different controller style did not change the cell area or the delay. Dali starts with an *initial FSM* which is one-hot encoded. After the optimization, the *final FSM* shows the min-encoded state encoding and the final number of states.

In all cases except the first one, the number of final states is lower than the number of initial states.

For the first three option combinations (see the first three lines in table 1), "Single Controller" is selected which produces the highest number of states and the largest cell area.

Selecting the "Partitioned Controller" option, the number of final states and the cell area is strongly reduced. Dali produces in this case two FSM's for the module "AHT_IN" with 8 and 53 final states each, if no other option is selected.

The last combination, with the selection of "Partitioned Controller" and "Use Counter" shows the least number of states, the least delay and a low cell area.

This combination is preferred for the synthesis of the five ASC modules.

4 Modeling five ASC-Modules with Dali

The following sections describe the modeling of five modules of the ASC with Dali:

1. The input module AHT_IN
2. The Header Translator module HT
3. The Shift Register Control module SRC
4. The Receive Data module RD

Options					Dali Results			
Single Contr.	Part. Contr.	Use Counter	Map Unr. States		Init. States	Final States	Cell Area	Delay (nsec)
*			*		478	512	3311	7.9
*		*	*		308	256	3131	5.63
*		*			308	196	3428	8.3
	*				478	8+53	1717	3.34
	*	*	*		308	8+32	1775	2.58
	*	*			308	8+29	1776	2.52

Table 1: Effects on option setting, synthesizing module AHT_IN

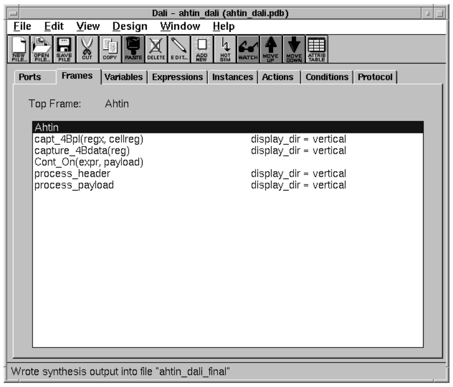
5. The AHT Output module AHT_Out

These modules are typical control oriented modules.
They do not contain arithmetic operations but data transmission and I/O functions like

1. Data synchronization with a strobe signal
2. Continuous data acceptance or data transmission
3. Data concatenation or splitting
4. Data replacement by data from look-up tables (e.g. the Routing Table)
5. Data encoding and decoding
6. One-way and two-way handshaking communication with other modules and processes.

4.1 Modeling of the ASC Input Module AHT_IN

The Address Header Translator Input module **AHT_IN**: accepts the ATM cells byte-parallel. The begin of a cell is marked by the signal "Cell_Sync_In". The incoming data bitwidth is changed from 8 bits to 32 bits by storing the incoming data one byte per clock cycle into 32 bit registers. With this method, the data processing inside the AHT can be reduced by a factor of four.



Top Frame: Ahtin

```
Ahtin
capt_4Bpl(regx, cellreg)              display_dir = vertical
capture_4Bdata(reg)                   display_dir = vertical
Cont_On(expr, payload)
process_header                        display_dir = vertical
process_payload                       display_dir = vertical
```

Wrote synthesis output into file "ahtin_dali_final"

Figure 6: Dali GUI with the frame tab opened

The first four bytes belong to the cell header and are transmitted to the Header Translator (HT). The fifth header byte, the Header Error Code (HEC) byte is ommitted, since it does not have any function in the ATM Layer.

The remainder of the cell, the "payload" with a length of 48 bytes is transmitted via a one-way handshaking communication to the first-in-first-out buffer CELL_FIFO.

AHT_IN is modeled as a two-process architecture as shown in figure 7. The first process processes the process header, the second one processes the payload.

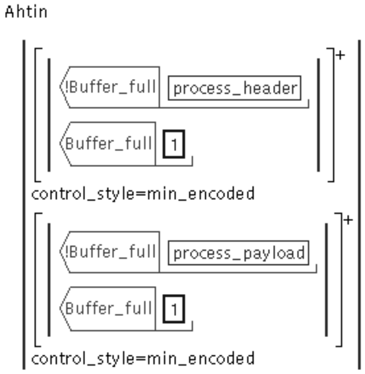


Figure 7: The top frame Ahtin

4.1.1 The AHT_IN Frame Hierarchy

Figure 6 shows the Dali GUI with “frames” tab selected. All frames of this module are listed.

The top frame is “Ahtin”, the subframes are “process_header” and “process_payload”. The “capture_4Bdata()” template-frame is a used as a subframe of “process_header” and the “capt_4Bpl()” template-frame is used as a subframe of “process_payload”.

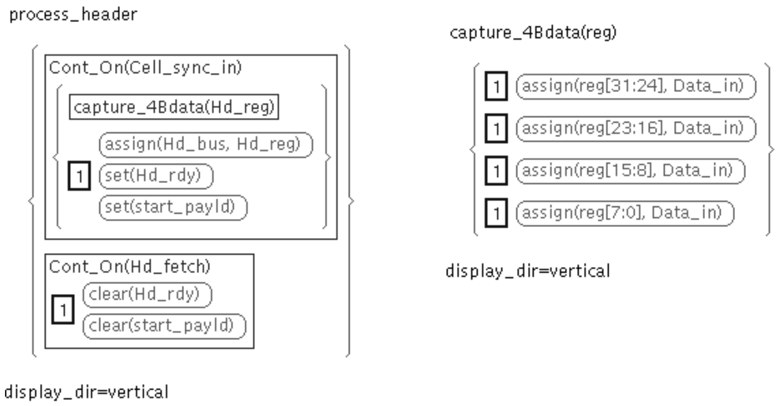


Figure 8: The Subframe “process_header”

Selecting the top frame “Ahtin”, the window figure 7 is displayed. It shows the two parallel executing processes:

- If the signal “Buffer_full” is not true, then execute the subframe process_header in an endless loop.

- If the signal “Buffer_full” is not true, then execute the subframe “process_payload” in an endless loop. The frame “process_payload” is triggered by the signal “start_payld”.

.

The first subframe “process_header”, together with the “capture_4Bdata()” template is shown in figure 8.

The logic is as follows: If the signal “Cell_sync_in” is active, then the 8 bit “Data_in” signal is read into the bit positions 31 to 24 of the 32 bit register “Hd_reg”. Sequentially with each clock cycle continuously the second, third and forth byte is read into the next downward bit positions of the register “Hd_reg”.

In the fifth clock cycle, the full Hd_reg is assigned to the signal “Hd_bus” and the “Hd_rdy” signal is set, which is the strobe signal for the Header Translator (HT) to accept the 4 byte header without the HEC byte. The HEC byte is not needed in the ATM Layer and therefore not accepted in the fifth clock cycle.

The process_payload frame is started with signal “start_payld” (refer to figure 9).

In order to reach continuous data acceptance, the payload data bytes are gathered alternatively in two 32 bit registers, “Cell_reg” and “Cell_reg2”, as shown in the frame

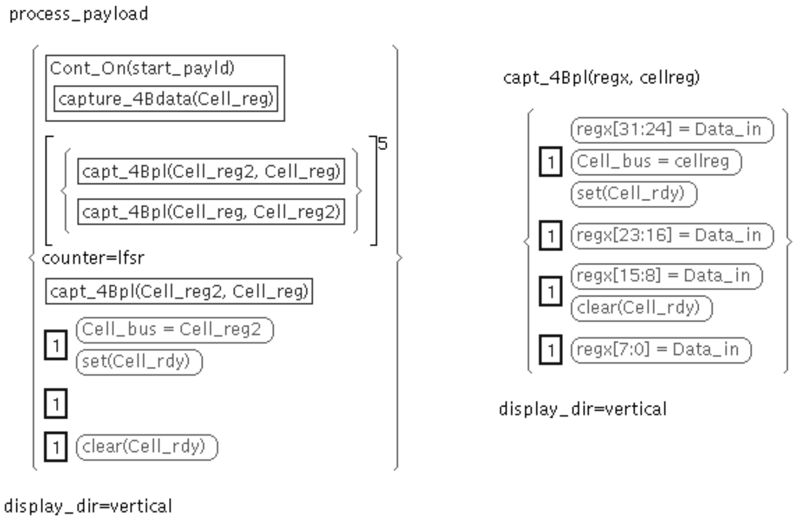


Figure 9: The process_payload frame

template capt_4Bpl(regx,cellreg) (see figure 9). The first 32 bit register (Cell_reg) is set to the cell bus one clock cycle after filling it with cell data. Cell_reg is stored in the CELL_FIFO buffer. In the mean time the second 32 bit register (Cell_reg2) is filled, and during its storage, the first register is used again. That means, the storing time of the registers in the CELL_FIFO must not exceed 4 clock cycles.

4.1.2 The AHTIN-FSM's

Dali produces 2 FSM's. The graphical representation can be displayed by the tool xvcg: (refer to figure 10).

4.1.3 Simulation of AHT_IN

The simulation of this design shows a correct behavior (see figure 11.)

4.1.4 Summary of Synthesis Results of Module AHT_IN

The Dali synthesis has been performed with the following parameters: (Release 1998.08)

Control Style: Binary-Min Ecoded.

Partitioned Controller.

Optimization Effort: high

Dali provides an optimization and delivers a controller report (see attachment). The following is reduced report which shows the estimated area and delay figures.

```
Controller Report
                              Initial    Final
                              -------    -----
```

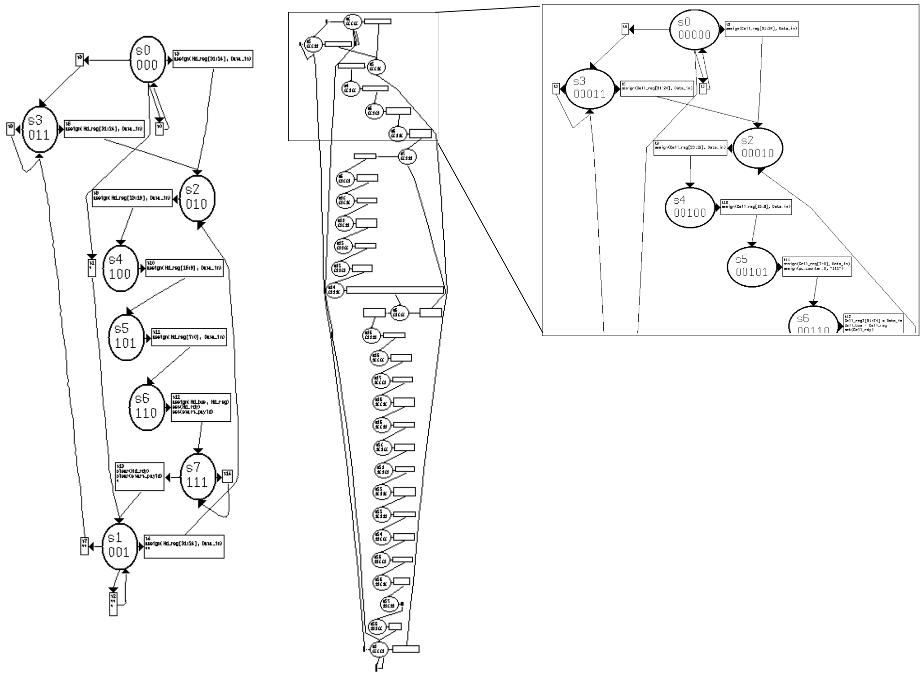


Figure 10: On the left hand side: FSM 1 of AHT_IN, on the right hand side: FSM 2 of AHT_IN, the upper six states are enlarged.

```
             States:       308    (8,29)
        Transitions:        --    (15,36)
    State Variables:        32    9=(3,5)+1

     Control Inputs:       178
    Action Triggers:        39

==============================================================================

Total combinational area   523.3
Total noncombinational    1253.0
Total area                1776.3

Delay through the combinational
part of the FSM             2.52 ns (    3.4 logic levels)

Library:
  7.00 gates per 1-bit D flipflop,
  1.00 gates per logical 2-bit NAND operation,
  0.75 ns per logic level
```

The final states (8,29) mean that two FSM’s were generated, the first one with 8 states, the second one with 29 states.

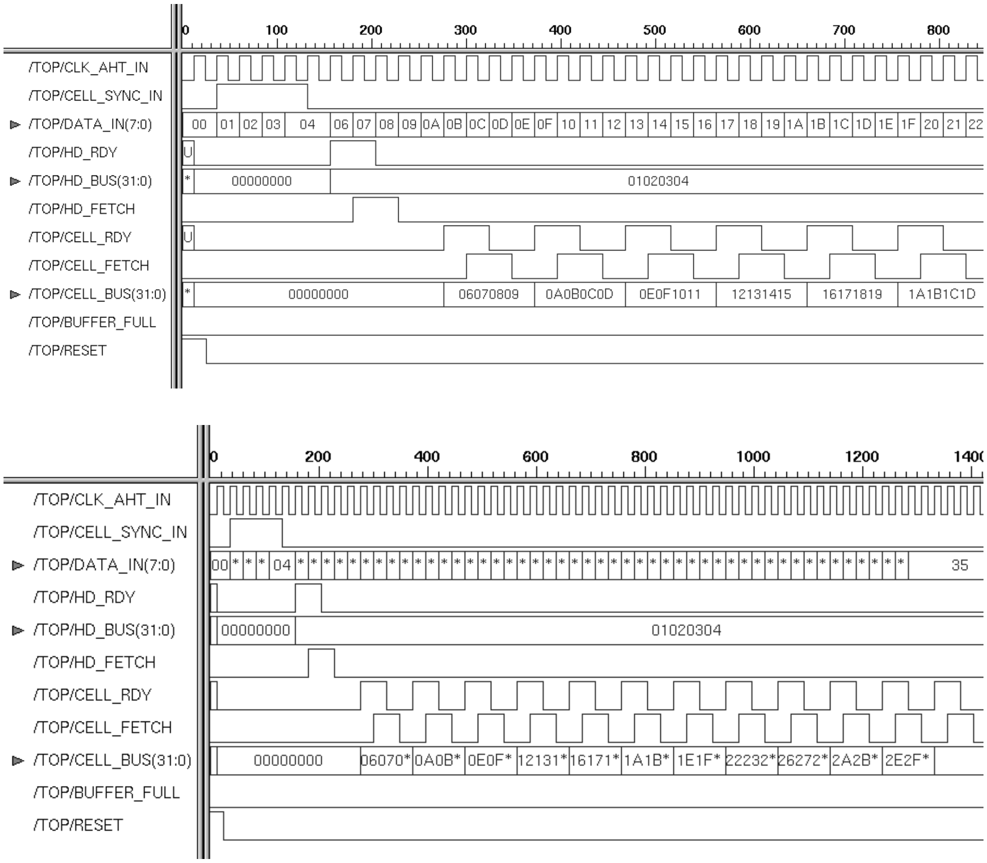


Figure 11: Simulation of AHT_IN, modeled with Dali. The upper diagram shows: The first bytes are transmitted. The lower diagram shows: One complete cell is transmitted. The upper scale in the simulation charts shows the time in nsec. The simulation clock cycle is set to 25 nsec.

4.2 Modeling of the Header Translator Module HT

The Header Translator module: (**HT**) receives the ATM-cell header from the input module AHT_IN via a two-way handshaking protocol. The strobe signal for the header is the signal "head_rdy". The header translator extracts the VPI/VCI-values from the header data, sets them into the variable "Address", and addresses the Routing table (RT). The RT sends new VPI/VCI values plus a routing tag "rtdata" back. The HT merges the RT data with some of the original header data and writes the new header plus the routing tag to the HEAD_FIFO buffer.

4.2.1 The HT frames

The Header Translator representation with Dali consists of the following three frames:

- HTIN is the main frame. It receives the header data and passes them to the next frame "addr_rt".

- addr_rt: addresses the Routing Table
- proc_rtdata: processes the Routing Table data and passes them to the HEAD_FIFO buffer.

The top frame is “HTIN” (refer to figure 12), it is framed by the Template Proc_On() which represents an endless process. It waits for the strobe signal “hd_rdy”, loads the header, sets the acknowledge signal hd_fetch, and waits until the strobe is deactivated. The address for the RT is composed of the VPI field concatenated to the VCI field. The next subframe “addr_rt” (refer to figure 12, right hand side) is called thereafter.

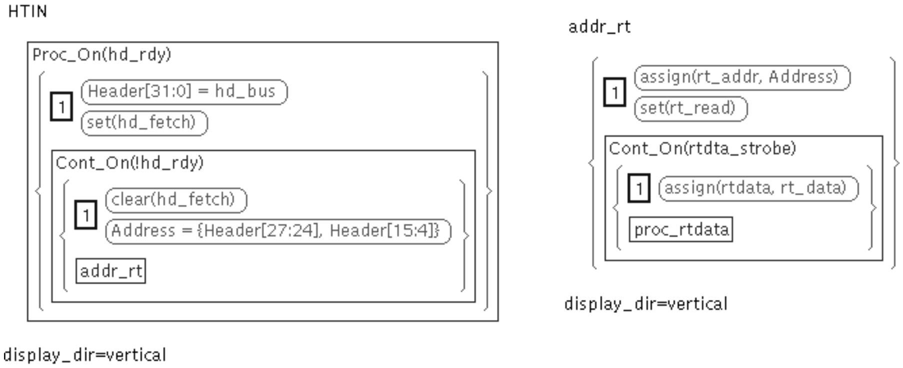


Figure 12: The top frame HTIN and the subframe add_rt

The subframe “addr_rt”, assigns the “Address”-Variable to the “rt_addr” signal and sets the “rt_rdy” signal for the Routing Table (RT) management. The RT management sets the signal “rtdta_strobe” if the RT data are available. The subframe “addr_rt” waits for the acknowledgment from the RT (rtdta_strobe), collects the RT data and processes them in the “proc_rtdata”-frame.

The “proc_rtdata-frame” (refer to figure 13 extracts the routing tag (rtag) and the VPI/VCI fields. It generates the new header, concatenates it with the routing tag and stores it to the HEAD_FIFO if the FIFO buffer is not full.

4.2.2 The HT FSM

The initial FSM is one-hot encoded and has 14 states (see figure 14, left hand side). The final FSM is binary-min encoded and has 8 states (see figure 14, right hand side).

4.2.3 Simulation of the Header Translator

The simulation of the Header Translator described with Dali shows a correct behavior (see figure 15.)

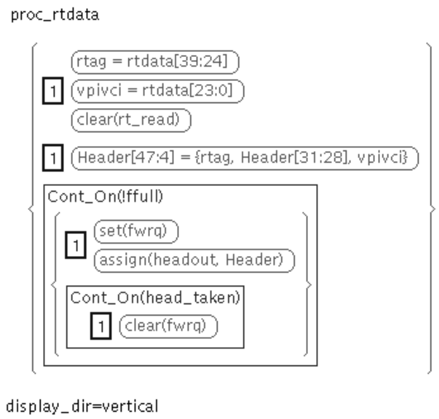


Figure 13: The proc_rtdata - frame

4.2.4 Synthesis Results of Module HT

Selected Options:

Control Style: Binary - Min Encoded
"Single Processor" architecture.
Optimization Effort: High
The following reduced report shows the estimated area and delay figures:

```
  Controller Report
                        Initial    Final
                        -------    -----

           States:         14        8
      Transitions:         25       13
  State Variables:         14        3

   Control Inputs:        288
  Action Triggers:         14

==============================================================================

Total combinational area   486.9
Total noncombinational    1519.0
Total area                2005.9

Delay through the combinational
part of the FSM             2.43 ns (    3.2 logic levels)

Library:
  7.00 gates per 1-bit D flipflop,
  1.00 gates per logical 2-bit NAND operation,
  0.75 ns per logic level
```

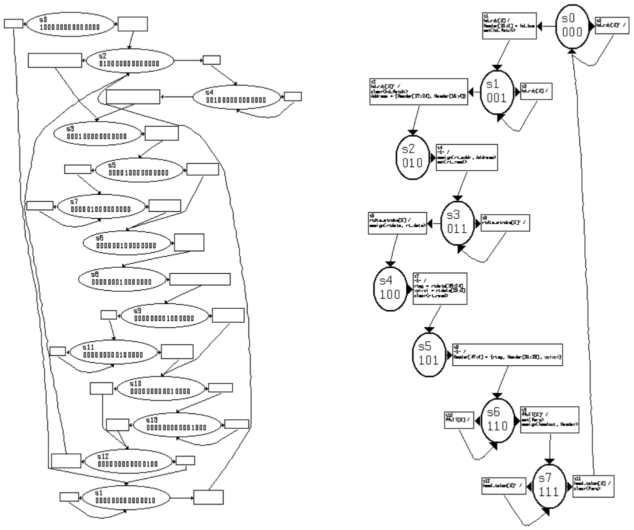


Figure 14: Dali FSM's. The left hand side shows the initial FSM. The right hand side shows the final FSM

4.3 The Shift Register Control Module (SRC)

The Shift Register Control module is described in a single process architecture. The SRC loads a shift register with 32 bits of cell data (not shown in the block diagram) to be shifted serially through the ATM Switch.

In order to recognize the start of the cell at the output side of the switch, the cell data is encoded by the 4B/5B encoding procedure, which is used in the "Taxi-Protocol". That means, a shift register load of 32 bits is encoded to 40 bits plus a start character of 5 bits, adding up to 45 bits shift register length. A complete cell transmission through the switch starts with a special cell start character plus 4 byte of header data (without Header Error Code (HEC)) encoded to 45 bits, followed by twelve shift register loads for the payload.

4.3.1 The SRCIN Frame

Figure 16 shows the top frame (SRCIN) of the SRC module. The "Proc_On()" is the template for the "endless process".

The first action is to load the cell header if the signal "load_head" is active. The "load_reg_encode" frame call encodes the register with header data. The shift register is ready if the "sr_ready" signal is active and the shifting can start if the "transmit signal" is active. The "Cont_on()" templates assure the correct handling of the two-way-handshaking protocol to the Routing Control module and to the shift register. The "load_shreg" frame call loads the shift register with the encoded data.

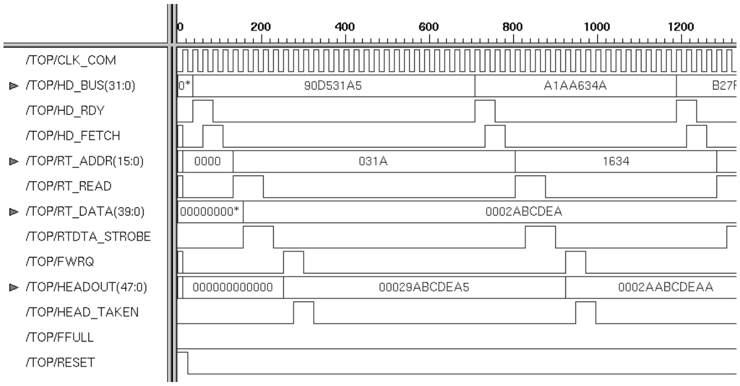


Figure 15: Simulation of HT, modeled with Dali.The upper scale in the simulation charts shows the time in nsec. The simulation clock cycle is set to 25 nsec.

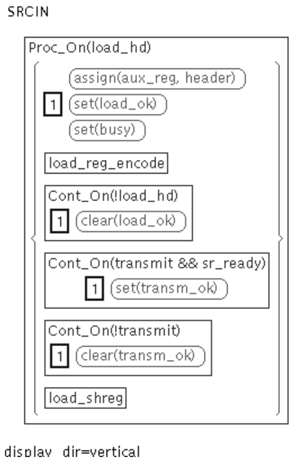


Figure 16: The SRCIN frame of the Shift Register Control Module

4.3.2 The “load_reg_encode” and the “load_shreg” Frames

The “load_reg_encode” frame (refer to figure 17) encodes one register load (32 bit) of the cell data with the 4B/5B encoding scheme. The encoding function “encd()” holds the encoding table and is located in the utilities package (utils_src.vhd). The result of the encoding is a 40 bit wide bit string.

The load_shreg frame (see figure 17, right hand side) sets the start-cell-character (“11001”) at the beginning of the first shift register load and passes it to the shift register with the “sr_strobe” signal.

4.3.3 The “shift_payload” Frame

The “shift_payload” frame (see figure 18) fetches the cell data out of the CELL_FIFO in register loads of 32 bit each, encodes them by calling the “load_reg_encode” frame, adds

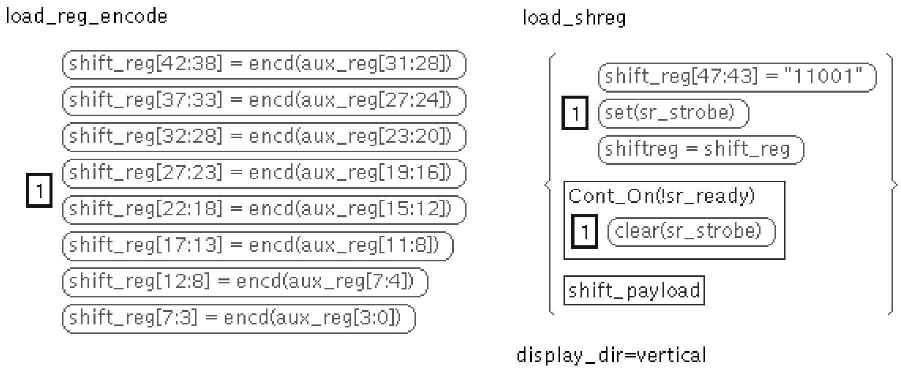

load_reg_encode
1
shift_reg[42:38] = encd(aux_reg[31:28])
shift_reg[37:33] = encd(aux_reg[27:24])
shift_reg[32:28] = encd(aux_reg[23:20])
shift_reg[27:23] = encd(aux_reg[19:16])
shift_reg[22:18] = encd(aux_reg[15:12])
shift_reg[17:13] = encd(aux_reg[11:8])
shift_reg[12:8] = encd(aux_reg[7:4])
shift_reg[7:3] = encd(aux_reg[3:0])
load_shreg
1
shift_reg[47:43] = "11001"
set(sr_strobe)
shiftreg = shift_reg
Cont_On(!sr_ready)
1
clear(sr_strobe)
shift_payload
display_dir=vertical

Figure 17: The load_reg_encode frame encodes the cell data and the load_shreg frame passes the the header data to the shift register

the cell-data-start-character ("10001") in front of each register load and passes the load to the shift register. Since the payload is 48 bytes long, this is performed in a loop which is repeated 12 times.

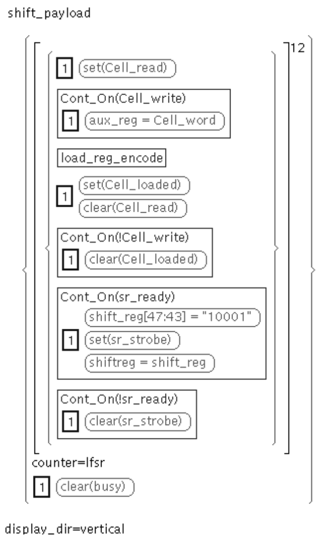


Figure 18: The load_shreg frame passes the header data to the shift register

4.3.4 The SRC-FSM

The graphical representation of the SRC-FSM is shown in figure 19.

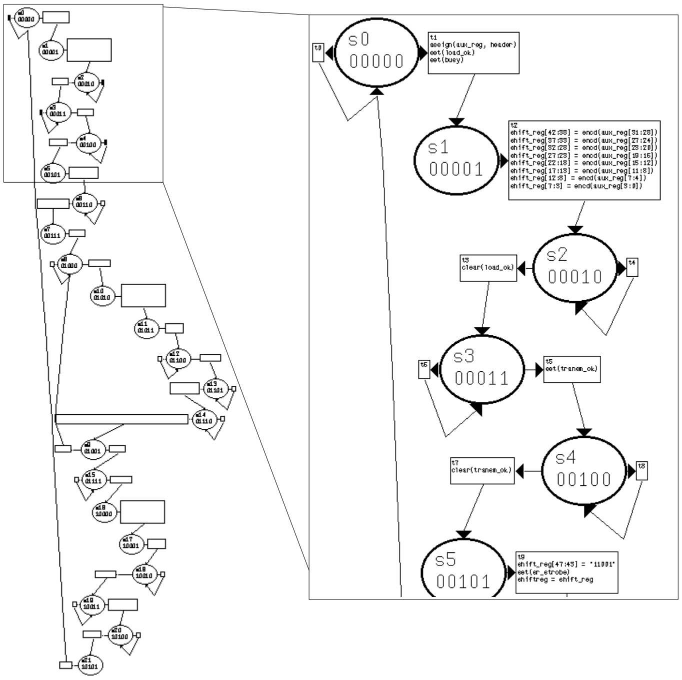

s0
00000
t0
t1
assign(aux_reg, header)
set(load_ok)
set(busy)
s1
00001
t2
shift_reg[42:38] = encd(aux_reg[31:28])
shift_reg[37:33] = encd(aux_reg[27:24])
shift_reg[32:28] = encd(aux_reg[23:20])
shift_reg[27:23] = encd(aux_reg[19:16])
shift_reg[22:18] = encd(aux_reg[15:12])
shift_reg[17:13] = encd(aux_reg[11:8])
shift_reg[12:8] = encd(aux_reg[7:4])
shift_reg[7:3] = encd(aux_reg[3:0])
t3
clear(load_ok)
s2
00010
t4
t6
s3
00011
t5
set(transm_ok)
t7
clear(transm_ok)
s4
00100
t8
s5
00101
t9
shift_reg[47:43] = "11001"
set(sr_strobe)
shiftreg = shift_reg

Figure 19: The SRC FSM. The right hand side box shows the upper six states enlarged.

4.3.5 Simulation of the Module SRC

Figure 20 shows the simulation results of the synthesized module SRC. The upper figure shows the cell header and the first cell words encoded and passed to the shift register. The lower figure shows how one complete cell is encoded and passed to the shift register.

4.3.6 Synthesis Results of Module SRC

The total report can be found in the attachment. The following is a summary of the synthesis figures.

```
          Protocol:  src
     Compile Style:  min_encoded
    Compile Effort:  high

  Controller Report
                       Initial    Final
                       -------    -----

            States:        48       22
       Transitions:        85       36
   State Variables:        26        5

    Control Inputs:       206
   Action Triggers:        38
```

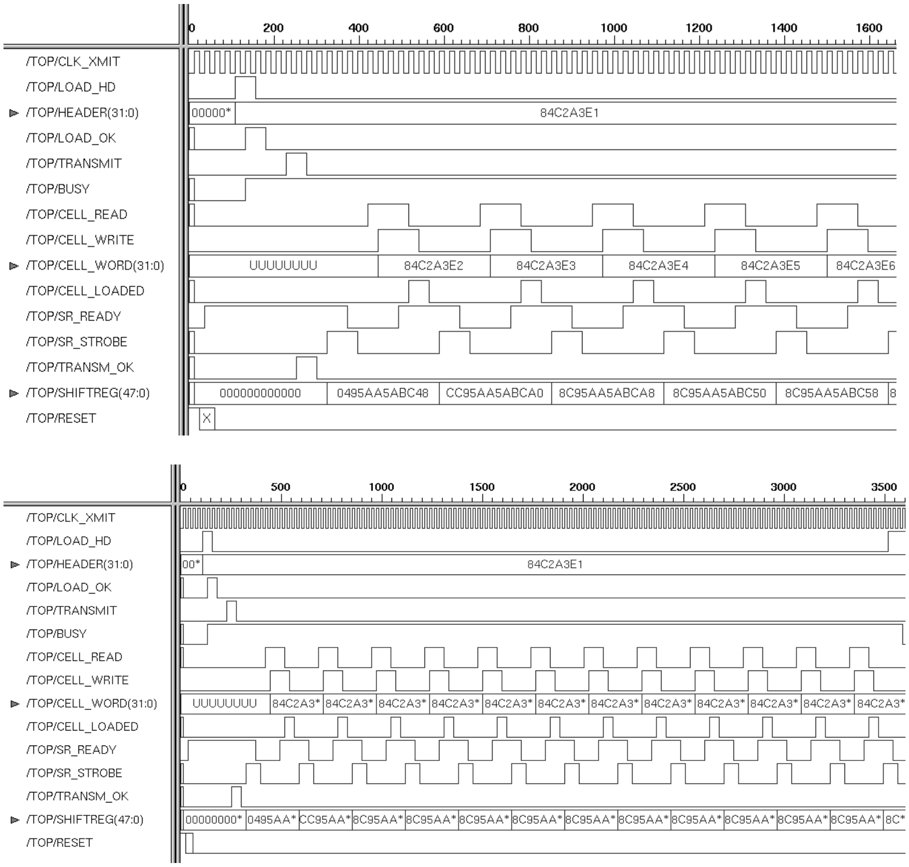


Figure 20: Simulation of module SRC: Upper figure: The cell header and the first cell words are passed encoded to the shift register. Lower figure: One complete cell is encoded and passed to the shift register.

```
Total combinational area   461.9
Total noncombinational    1043.0
Total area                1504.9

Delay through the combinational
part of the FSM             3.28 ns (    4.4 logic levels)

Library:
  7.00 gates per 1-bit D flipflop,
  1.00 gates per logical 2-bit NAND operation,
  0.75 ns per logic level
```

4.4 The Receive Data Module (RD)

The Receive Data module accepts data from a shift register (not shown in the block diagram) at the output of the ATM switch and stores them in the FIFO_Out buffer.

The Receive Data module is described in two processes. The first process takes the

4B/5B encoded data from the shift register and decodes them to 4 bit data. The second process, running in parallel, stores the data in the FIFO_Out buffer.

4.4.1 The RDIN Frame

Figure 21 shows the top frame (RDIN) of the RD module. Two endless running processes are shown. Both processes accept if the FIFO_Out buffer is not full.

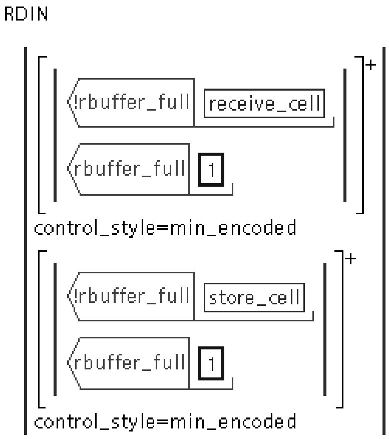


Figure 21: The RDIN frame of the Receive Data Module

IF the first process accepts, the “receive_cell” frame is executed. IF the second process accepts, the “store_cell” frame is executed.

4.4.2 The “receive_cell” Frame

The “receive_cell” frame (refer to figure 22) accepts the cell data from the shift register at the output side of the ATM switch. The “Cont_On()” template frame waits for the shift register strobe “sr_data” and takes the encoded data into a 48-bit register. The acknowledge signal “srdta_taken” is raised.

If the first 5 bit of the cell data are equal to the “cell-start” special character (“11001”), the first register load (40 bit) is decoded in the “decode_register” frame and passed to the “store_register” frame thereafter.

If the first 5 bit of the shift register load contain the shift register start special character(“10001”), the “decode_register” frame (refer to figure 22) is called and the store_register frame is executed thereafter. This sequence is repeated twelve times, to accept the 48 bytes of the payload. Unrolling the loop would lead to a higher area, therefore the loop is kept rolled and a loop counter is implemented.

4.4.3 The “store_cell” Frame

The “store_cell frame” (see figure 23) stores one register load (32 bits) into the FIFO_Out buffer, using the two-way-handshaking protocol.

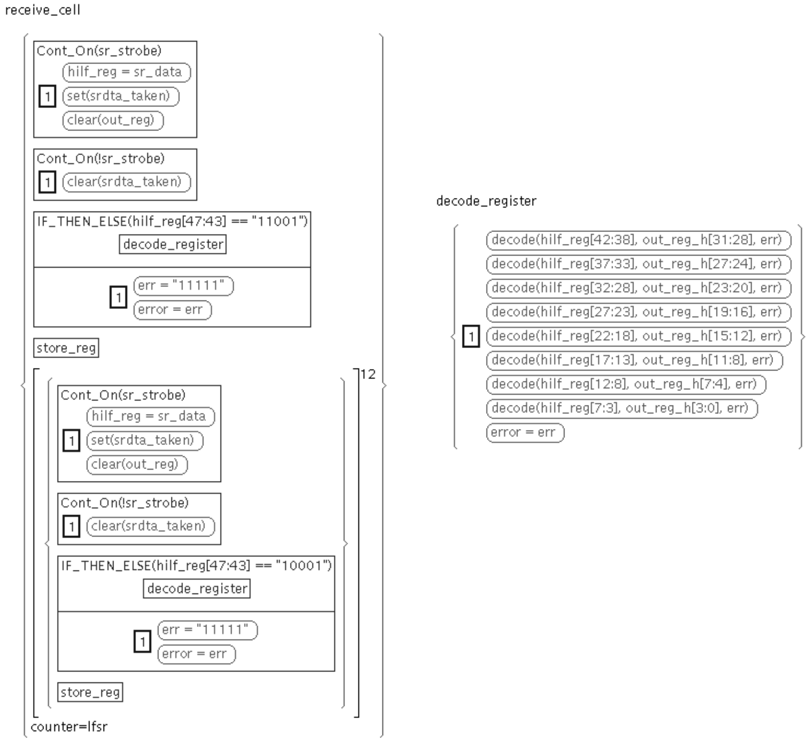

receive_cell
Cont_On(sr_strobe)
hilf_reg = sr_data
1
set(srdta_taken)
clear(out_reg)
Cont_On(!sr_strobe)
1
clear(srdta_taken)
IF_THEN_ELSE(hilf_reg[47:43] == "11001")
decode_register
1
err = "11111"
error = err
store_reg
12
Cont_On(sr_strobe)
hilf_reg = sr_data
1
set(srdta_taken)
clear(out_reg)
Cont_On(!sr_strobe)
1
clear(srdta_taken)
IF_THEN_ELSE(hilf_reg[47:43] == "10001")
decode_register
1
err = "11111"
error = err
store_reg
counter=lfsr
decode_register
decode(hilf_reg[42:38], out_reg_h[31:28], err)
decode(hilf_reg[37:33], out_reg_h[27:24], err)
decode(hilf_reg[32:28], out_reg_h[23:20], err)
decode(hilf_reg[27:23], out_reg_h[19:16], err)
1
decode(hilf_reg[22:18], out_reg_h[15:12], err)
decode(hilf_reg[17:13], out_reg_h[11:8], err)
decode(hilf_reg[12:8], out_reg_h[7:4], err)
decode(hilf_reg[7:3], out_reg_h[3:0], err)
error = err

Figure 22: The receive_cell frame decodes the cell data and the decode_registerframe decodes one register load

4.4.4 The “decode_reg” Frame

The actions in the “decode_reg” frame (refer to figure 22) call the decode procedure in the utilities package (utilsrd.vhd).

The decode procedure holds the decoding table of the 5B/4B decoding scheme. The input to the decoding procedure is the 40 bit register load (without start character) and the result of the decoding is a 32 bit wide bit string. An error is returned, if a bit pattern is not found in the table.

4.4.5 The “store_reg” Frame

The “store_reg” frame (see figure 23) passes one decoded register load (32 bits) to the second process.

4.4.6 The FSM Graphs

Figure 24 shows the two FSM’s generated by Dali in graphical form, displayed by the xvcg program.

The FSM on the left hand side in figure 24 controls the “receive_cell” frame, the FSM on the right hand side controls the “store_cell” frame.

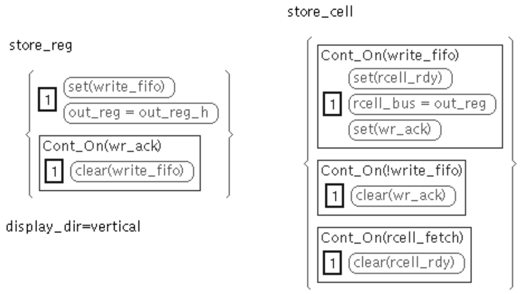


Figure 23: The store_reg frame passes one decoded register load to the second process and the store_cell frame stores one register load (32 bits) into the FIFO_Out buffer

4.4.7 Simulation of the Module RD

Dali provides a testdriver template which has to be completed with appropriate stimuli for the simulation.

The figures 25 and 26 show the simulation results of the synthetsized module RD. The upper figure shows the cell header and the first cell words decoded and passed to the shift register (see figure 25). The sr_data signal (48 bits) shows the encoded data with start character. The rcell_bus signal (32 bits) shows the decoded data without start character.

The second figure (see figure 26) shows one complete cell accepted and passed to the FIFO_Out buffer.

4.4.8 Synthesis Results of Module RD

The total report can be found in the attachment. The following is a summary of the synthesis figures.

```
           Protocol:  rd
      Compile Style:  automatic
     Compile Effort:  high

==============================================================================

  Controller Report
                        Initial    Final
                        -------    -----

             States:       261    (18,5)
        Transitions:        --    (38,13)
    State Variables:        28    9=(5,3)+1

     Control Inputs:       245
    Action Triggers:        43

==============================================================================
```

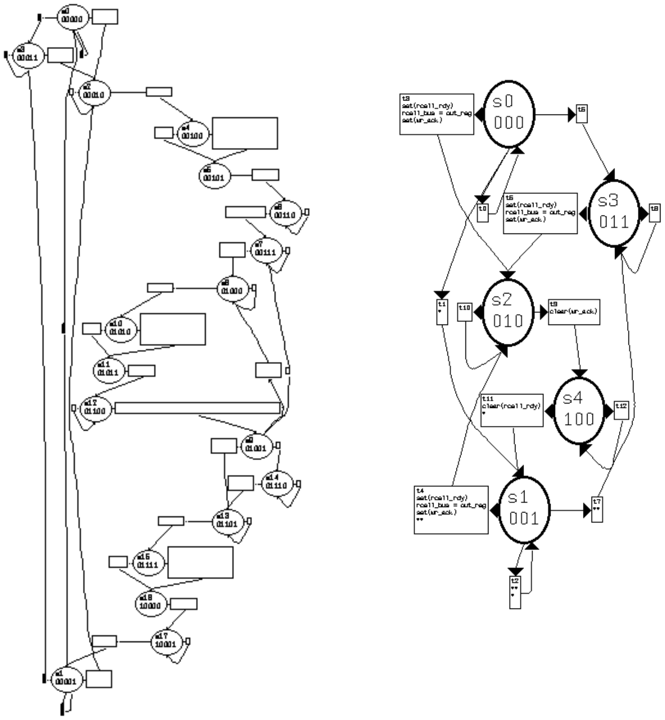


Figure 24: On the left hand side: FSM controling the “receive_cell” frame, on the right hand side: FSM controling the “store_cell” frame.

```
Estimated size of protocol:
FSM
  combinational area       184.6
  noncombinational area     70.0  (10 bits)
  total area               254.6
Counters required for repeat frames
  combinational area        19.0  (assuming no sharing of decrementers)
  noncombinational area     28.0  (4 bits total)
  total area                47.0

Variables and ports
  combinational area       388.0  (reset logic)
  noncombinational area   1358.0  (194 bits total)
  total area              1746.0

Combinational logic of
actions is NOT included
                         =======
Total combinational area   591.6
Total noncombinational    1456.0
Total area                2047.6

Delay through the combinational
```

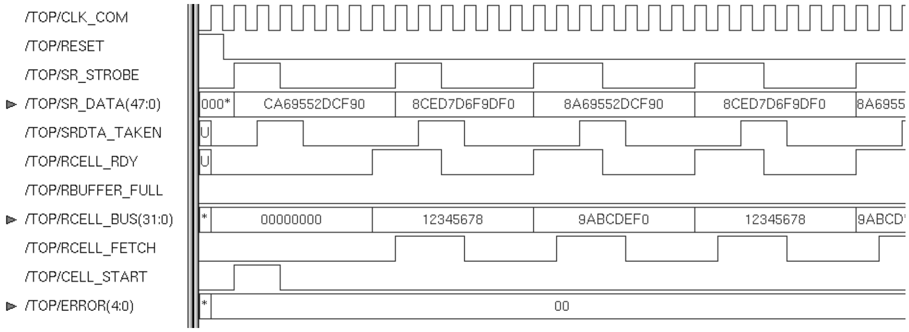


Figure 25: Simulation of module RD: The cell header and the first cell words are decoded to the shift register.

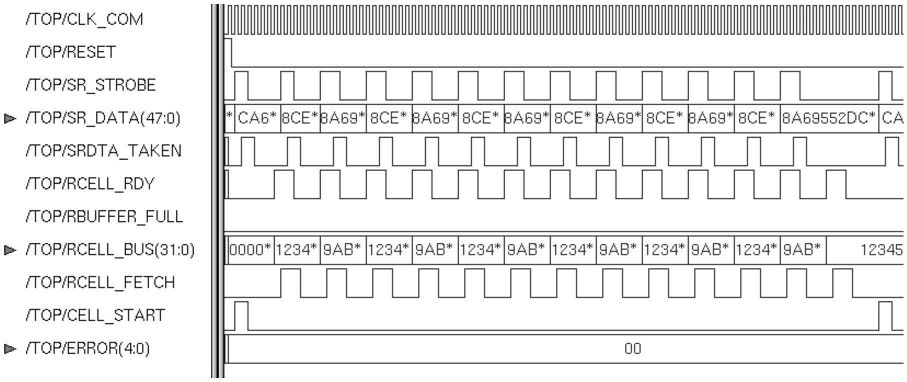


Figure 26: One complete cell is accepted and passed to the FIFO_Out buffer.

```
part of the FSM             3.53 ns (    4.7 logic levels)
```

4.5 The AHT Output Module (AHT_OUT)

The AHT output module AHT_OUT fetches the data from the FIFO_Out buffer and sets them at the output line, 8 bits per clock cycle, if the output channel is free.

The AHT output module is described in two processes. The first process fetches the data from the FIFO_Out buffer in portions of 32 bits and passes them to the second process. The second process puts the cell data consecutively - without data gap - 8 bit parallel per clock cycle at the output line, if the output channel is free.

4.5.1 The AHTOUT Frame

Figure 27 shows the top frame (AHTOUT) of the AHT_OUT module. Two endless running processes are shown. Both processes accept if the signal “chan_bsy” is not active.

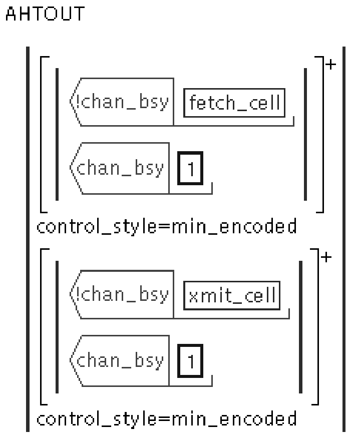


Figure 27: The AHTOUT frame.

IF the first process accepts, the “fetch_cell” frame is executed. IF the second process accepts, the “xmit_cell” frame is executed.

4.5.2 The “fetch_cell” Frame

The “fetch_cell frame” (refer to figure 28) fetches the cell data via a two-way handshake protocol from the FIFO_Out buffer manager. The “read_cell” signal alerts the FIFO_Out manager to set 32 bits of cell data at the “rcell_data” line.

The “fetch_cell” frame passes the cell data to the xmit_cell frame.

It is important, that the two-way handshake protocol does not take more than 4 clock cycles of the “Clk_aht_out” to transmit the 4 bytes of cell data.

4.5.3 The “xmit_cell” Frame

The “xmit_cell” frame (refer to figure 28) receives a “transmit” signal from the first process in case cell data is available. It sends a “cell_presync” signal out, one clock cycle ahead of the cell data. The first cell byte is set on the output data line together with the “cell_sync_out” signal. The “xmit_cell” frame partitions one 32 bit register load of cell data into parts of 8 bit and puts them consecutively on the output line, one byte per clock cycle.

4.5.4 The “xmit_4bytes” Frame

The “xmit_4bytes” frame (see figure 28) is a subframe of the “xmit_cell” frame and passes one register load (32 bits) to the “data_out” line.

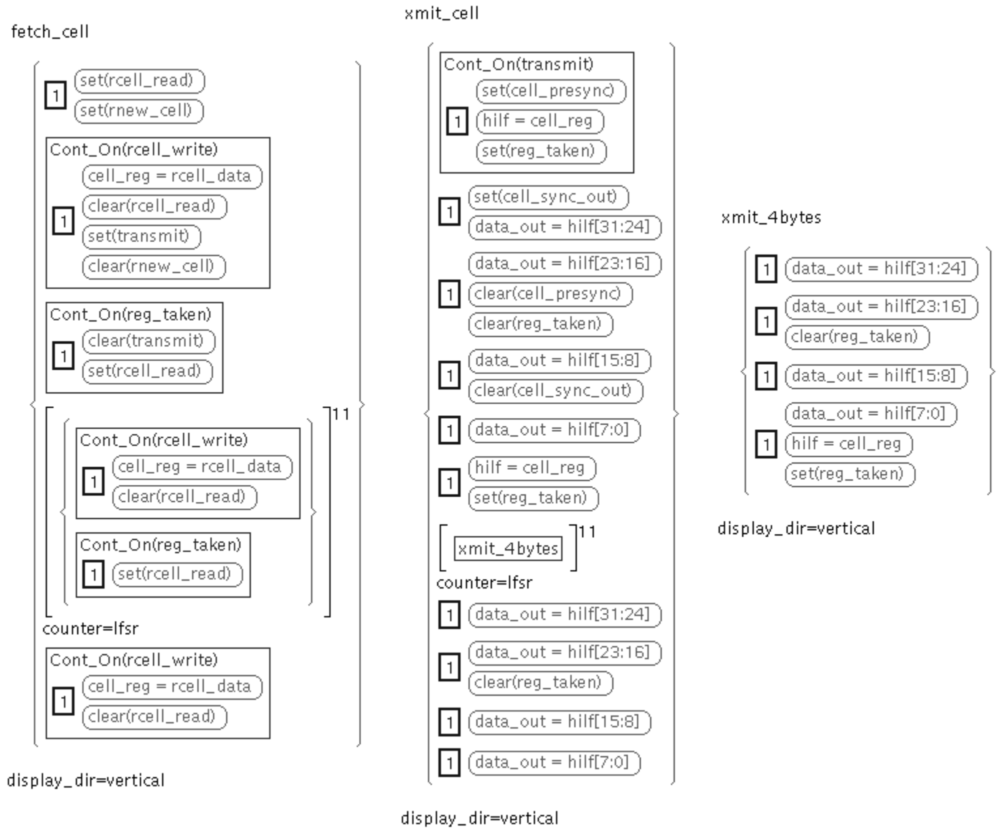


Figure 28: The fetch_cell frame fetches the cell data from the FIFO_Out buffer, the “xmit_cell” frame puts one byte each per clock cycle on the output line and the xmit_4bytes frame as a subframe of “xmit_cell” sets 4 bytes on the data_out line

4.5.5 The FSM Graphs

Figure 29 shows the two FSM’s generated by Dali in graphical form, displayed by the xvcg program.

The FSM on the left hand side controls the “fetch_cell” frame. The FSM on the right hand side of figure 29 controls the “xmit_cell” frame.

4.5.6 Simulation of the Module AHT_Out

Dali provides a testdriver template which has to be completed with appropriate stimuli for the simulation.

The following figures show the simulation results of the synthesized module AHT_Out.

The upper figure shows the cell header and the first cell words fetched from the FIFO_Out buffer and passed to the output line (see figure 30).

The second figure (see figure 31) shows one complete cell passed to the output line.

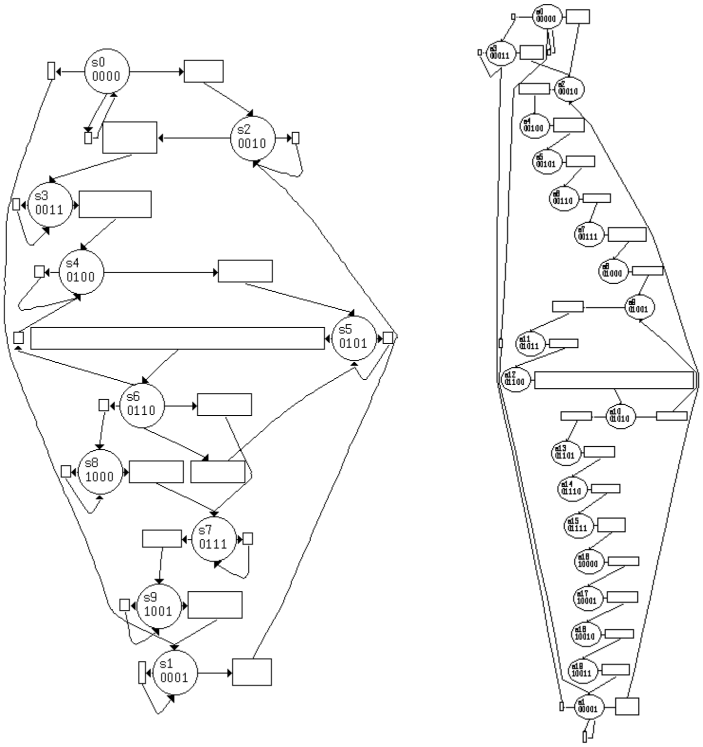


Figure 29: On the left hand side: FSM of the “fetch_cell” frame. On the right hand side: FSM of the “xmit_cell” frame.

4.5.7 Synthesis Results of Module AHT_Out

The total report can be found in the attachment. The following is a summary of the synthesis figures.

```
  Controller Report
                      Initial    Final
                      -------    -----
           States:       543    (10,20)
      Transitions:        --    (23,27)
    State Variables:      31    10=(4,5)+1

     Control Inputs:     120
    Action Triggers:      42
==============================================================================
Total combinational area   402.5
Total noncombinational     735.0
Total area                1137.5

Delay through the combinational
part of the FSM             2.61 ns (    3.5 logic levels)
```

%enddocument

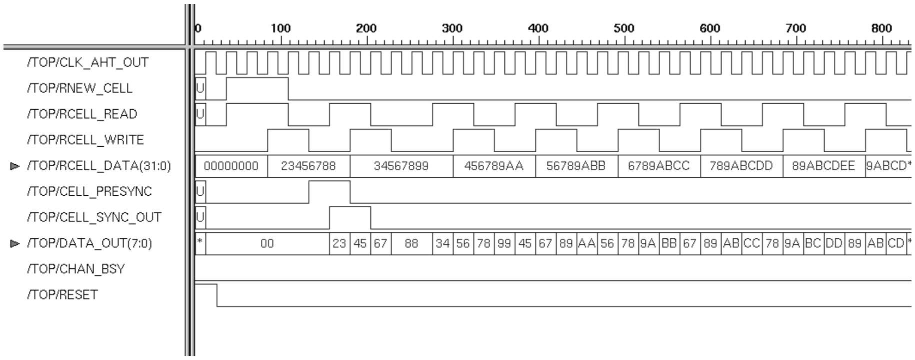


Figure 30: Simulation of module AHT_Out: The cell header and the first cell words are fetched from the FIFO_Out buffer and passed to the output line.

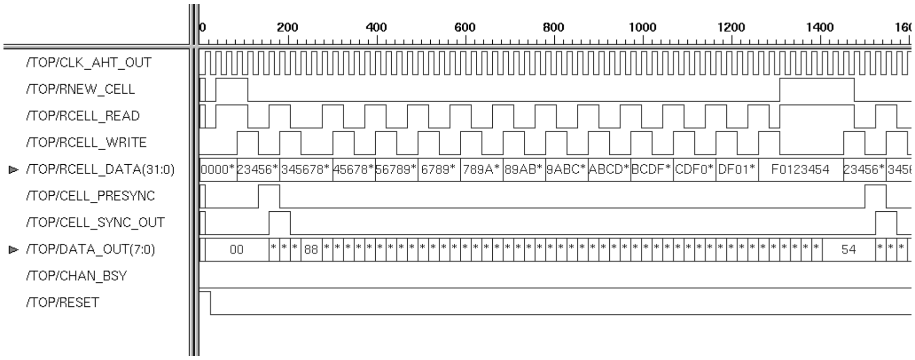


Figure 31: One complete cell is passed to the output line.

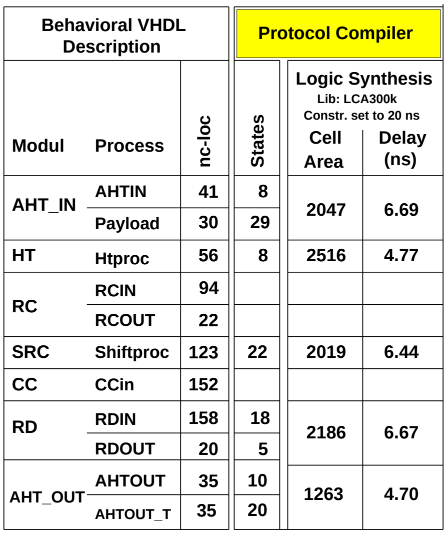
5 Summary of the Results

The results of the PC$^{\mathrm{TM}}$ synthesis for the modules AHT_IN, HT, SRC, RD and AHT_OUT and the appropriate processes are shown on the right hand side of the table 2. The states of the processes multiplied with the clock cycle time are a measure for the execution time of the process, not counting the loops. One of the PC$^{\mathrm{TM}}$ outputs is a RT-VHDL description which is simulated (see above) and fed to the Design Compiler$^{\mathrm{TM}}$, in order to perform a Logic Synthesis and optimization. The results of the Logic Synthesis plus optimization regarding the cell area and delay are shown. As target technology library a “logic cell array” library (LCA300k) is used. The time constraint for the clock cycle is set to 20 ns.

The following options are set for the PC synthesis:

- Control style: Binary-min encoded
- Architecture used: “Partitioned Controller” for the following modules: AHT_IN, RD, AHT_OUT.

  “Single Controller” for the modules HT and SRC.



Behavioral VHDL Description			Protocol Compiler		
				Logic Synthesis Lib: LCA300k Constr. set to 20 ns	
Modul	Process	nc-loc	States	Cell Area	Delay (ns)
AHT_IN	AHTIN	41	8	2047	6.69
	Payload	30	29		
HT	Htproc	56	8	2516	4.77
RC	RCIN	94			
	RCOUT	22			
SRC	Shiftproc	123	22	2019	6.44
CC	CCin	152			
RD	RDIN	158	18	2186	6.67
	RDOUT	20	5		
AHT_OUT	AHTOUT	35	10	1263	4.70
	AHTOUT_T	35	20		

Table 2: Synthesis results of five ASC modules using the Protocol Compiler

- “Use Counters” for all loops with a fixed number of loop traversals.

The PC version 1998.08 is used. The run time for synthesis and code generation performed on a SUN^{T}M Ultra 5 is a few seconds for each of the above five modules.

The descriptions for the modules are graphically generated in symbolic form. The PC delivers an output for all modules.

6 Conclusion

The Protocol Compiler from Synopsys^{TM} is used to describe the behavior of five control oriented ATM Switch Controller modules. The advantage of this graphical representation method is, that a design can be described fast, compact and well comprehensible allowing short design cycles for data transmission and protocol logic circuits.

7 Acknowledgment

Thanks to Ulli Holtmann from Synopsys^{TM} Corp., who gave me many valuable hints how to use the Protocol Compiler.

8 Attachment: PC Synthesis Reports

8.1 Input Module AHT_IN

8.1.1 The Dali Synthesis Report for AHT_IN

```
Processing /afs/informatik.uni-tuebingen.de/home/wlange/hilan/aht/ahtin_dali/ahtin.pdb
Information: Removing prior stored synthesis information from ahtin.pdb (PCC-9)
Creating initial circuit
Inserting LFSR for repeat operator count=5
Optimizing logic structure
style = automatic
Analyzing Protocol
shell=0 states=1 (controller states)
shell=1 states=6 (controller states)
shell=2 states=11 (controller states)
optimizing...
shell=3 states=18 (controller states)
shell=4 states=27 (controller states)
shell=5 states=38 (controller states)
shell=6 states=58 (controller states)
shell=7 states=74 (controller states)
shell=8 states=83 (controller states)
shell=9 states=92 (controller states)
shell=10 states=101 (controller states)
shell=11 states=110 (controller states)
shell=12 states=119 (controller states)
shell=13 states=128 (controller states)
shell=14 states=137 (controller states)
shell=15 states=146 (controller states)
shell=16 states=155 (controller states)
shell=17 states=164 (controller states)
shell=18 states=173 (controller states)
shell=19 states=182 (controller states)
shell=20 states=191 (controller states)
shell=21 states=200 (controller states)
shell=22 states=209 (controller states)
shell=23 states=218 (controller states)
shell=24 states=227 (controller states)
shell=25 states=236 (controller states)
shell=26 states=245 (controller states)
shell=27 states=254 (controller states)
shell=28 states=279 (controller states)
shell=29 states=290 (controller states)
shell=30 states=299 (controller states)
shell=31 states=308 (controller states)
--------------------------------------------------------------------------
: node :   states  :  state vars  : name                                 :
--------------------------------------------------------------------------
:    0 :       308 : [   1:31  ]  : Ahtin
:    1 :       308 : [   1:31  ]  : alternative
:    2 :        10 : [   1:9   ]  : repeat+
:    3 :        10 : [   1:9   ]  : alternative
:    4 :         9 : [   1:8   ]  : if
:    5 :         9 : [   1:8   ]  : process_header
:    6 :         9 : [   1:8   ]  : process_header
:    7 :         9 : [   1:8   ]  : sequence
:    8 :         7 : [   1:6   ]  : Cont_On
:   12 :         7 : [   1:6   ]  : Cont_On
:   13 :         7 : [   1:6   ]  : sequence
:   16 :         6 : [   2:6   ]  : if
:   17 :         6 : [   2:6   ]  : sequence
:   18 :         6 : [   2:6   ]  : sequence
:   19 :         5 : [   2:5   ]  : capture_4Bdata
:   20 :         5 : [   2:5   ]  : capture_4Bdata
:   21 :         5 : [   2:5   ]  : sequence
:   27 :         3 : [   7:8   ]  : Cont_On
:   29 :         3 : [   7:8   ]  : Cont_On
:   30 :         3 : [   7:8   ]  : sequence
:   38 :        36 : [  10:31  ]  : repeat+
:   39 :        36 : [  10:31  ]  : alternative
:   40 :        35 : [  10:30  ]  : if
:   41 :        35 : [  10:30  ]  : process_payload
:   42 :        35 : [  10:30  ]  : process_payload
:   43 :        35 : [  10:30  ]  : sequence
:   44 :         6 : [  10:14  ]  : Cont_On
:   46 :         6 : [  10:14  ]  : Cont_On
:   47 :         6 : [  10:14  ]  : sequence
:   50 :         5 : [  11:14  ]  : if
:   51 :         5 : [  11:14  ]  : sequence
:   52 :         5 : [  11:14  ]  : capture_4Bdata
:   53 :         5 : [  11:14  ]  : capture_4Bdata
:   54 :         5 : [  11:14  ]  : sequence
:   59 :        18 : [  15:23  ]  : repeat
:   60 :         9 : [  15:22  ]  : sequence
:   61 :         5 : [  15:18  ]  : capt_4Bpl
:   62 :         5 : [  15:18  ]  : capt_4Bpl
:   63 :         5 : [  15:18  ]  : sequence
:   68 :         5 : [  19:22  ]  : capt_4Bpl
:   69 :         5 : [  19:22  ]  : capt_4Bpl
:   70 :         5 : [  19:22  ]  : sequence
:   75 :         5 : [  24:27  ]  : capt_4Bpl
:   76 :         5 : [  24:27  ]  : capt_4Bpl
:   77 :         5 : [  24:27  ]  : sequence
--------------------------------------------------------------------------
Partitioning node    2 : states=  10 : Frame Ahtin, repeat ’+’. <ahtin_dali-Ahtin-3>
Partitioning node   38 : states=  36 : Frame Ahtin, repeat ’+’. <ahtin_dali-Ahtin-9>
Compiling partitions
------------------------------------------------------------------------------
Optimizing /2
------------------------------------------------------------------------------
Extracting ahtin_dali_part_/2
Initial (9 state vars, 9 actions)
10 states
Creating state graph for ahtin_dali_part_/2
Simplifing transitions
Simplified 10 transition(s)
Initial (10 states, 20 transitions)
Minimizing state graph ahtin_dali_part_/2
Final (8 states, 15 transitions)
Encoding using minimum bits and binary codes
Converting to circuit REDUCED_ahtin_dali_part_/2
Simplifying logic equations.
Using CASE statement action block: 9 actions 6 triggers used, 6 cases
Merging VC_MINENC_REDUCED_ahtin_dali_part_/2
Final (3 state vars, 10 actions)
------------------------------------------------------------------------------
Optimizing /38
------------------------------------------------------------------------------
Extracting ahtin_dali_part_/38
Initial (22 state vars, 30 actions)
36 states
Creating state graph for ahtin_dali_part_/38
Simplifing transitions
Simplified 35 transition(s)
Initial (36 states, 48 transitions)
Minimizing state graph ahtin_dali_part_/38
Final (29 states, 36 transitions)
Encoding using minimum bits and binary codes
Converting to circuit REDUCED_ahtin_dali_part_/38
Simplifying logic equations.
Simplified 8 logic functions
Using CASE statement action block: 30 actions 20 triggers used, 19 cases
Merging VC_MINENC_REDUCED_ahtin_dali_part_/38
Final (5 state vars, 31 actions)
-----------------------------------------------------------------------------
Final wiring of ahtin_dali_part
-----------------------------------------------------------------------------
Top level controller implemented in distributed style using 1 state var
-----------------------------------------------------------------------------

==============================================================================

         Input File:  /afs/informatik.uni-tuebingen.de/home/wlange/hilan/aht/ahtin_dali/ahtin.pdb
           Protocol:  ahtin_dali
      Compile Style:  automatic
     Compile Effort:  high

==============================================================================

  Controller Report
                        Initial    Final
                        -------    -----

             States:       308    (8,29)
        Transitions:        --    (15,36)
    State Variables:        32    9=(3,5)+1

     Control Inputs:       178
    Action Triggers:        39

==============================================================================
Cleaning up
Optimizing logic structure
Estimating control logic...
Network optimization of control logic
                      area  delay
                      [ND2] [logic leves]
             initial    426  5.8
               sweep    238  5.0
     structure(area)    160  3.5
      structure(tds)    243  6.7
              redund    196  4.9
               sweep    185  4.7
     structure(area)    157  3.4
               sweep    157  3.4

Estimated size of protocol:
FSM
  combinational area       177.3
  noncombinational area     70.0  (10 bits)
  total area               247.3
```

```
Counters required for repeat frames
  combinational area        14.0  (assuming no sharing of decrementers)
  noncombinational area     21.0  (3 bits total)
  total area                35.0

Variables and ports
  combinational area       332.0  (reset logic)
  noncombinational area   1162.0  (166 bits total)
  total area              1494.0

Combinational logic of
actions is NOT included
                         =======
Total combinational area   523.3
Total noncombinational    1253.0
Total area                1776.3

Delay through the combinational
part of the FSM             2.52 ns (    3.4 logic levels)

Library:
  7.00 gates per 1-bit D flipflop,
  1.00 gates per logical 2-bit NAND operation,
  0.75 ns per logic level
Storing data
Writing ahtin.pdb+ (synthesized pdb)
```

8.1.2 The Logic Synthesis Report for AHT_IN

```
                      Behavioral Compiler (TM)
                        DC Professional (TM)
                           DC Expert (TM)
                         FPGA Compiler (TM)
                         VHDL Compiler (TM)
                          HDL Compiler (TM)
                        Library Compiler (TM)
                         Power Compiler (TM)
                         Test Compiler (TM)
                       Test Compiler Plus (TM)
                            CTV-Interface
                          ECO Compiler (TM)
                          DesignPower (TM)

                   Version 1998.08 -- Jun 25, 1998
              Copyright (c) 1988-1998 by Synopsys, Inc.
                         ALL RIGHTS RESERVED

This program is proprietary and confidential information of Synopsys, Inc.
and may be used and disclosed only as authorized in a license agreement
controlling such use and disclosure.

Initializing...
/* dc_script fuer rtl-eingangskomponente  - ahtin_rtl.vhd...  */

analyze -f vhdl ht_syn.vhd
Loading db file '/afs/informatik.uni-tuebingen.de/ti/synopsys/1998.8/libraries/syn/standard.sldb'
Loading db file '/afs/informatik.uni-tuebingen.de/ti/synopsys/1998.8/libraries/syn/gtech.db'
Loading db file '/afs/informatik.uni-tuebingen.de/ti/synopsys/1998.8/libraries/syn/lca300kv.db'
Reading in the Synopsys vhdl primitives.
/afs/informatik.uni-tuebingen.de/home/wlange/hilan/aht/ht_dalils/ht_syn.vhd:
1

elaborate ht

Inferred memory devices in process 'PC_PROC'
in routine ht line 142 in file
         '/afs/informatik.uni-tuebingen.de/home/wlange/hilan/aht/ht_dalils/ht_syn.vhd'.
===============================================================================
|    Register Name    |   Type    | Width | Bus | MB | AR | AS | SR | SS | ST |
===============================================================================
|     Address_reg     | Flip-flop |  16   |  Y  | N  | N  | N  | N  | N  | N  |
|     Header_reg      | Flip-flop |  48   |  N  | N  | ?  | ?  | ?  | ?  | ?  |
|      fwrq_reg       | Flip-flop |   1   |  -  | -  | N  | N  | N  | N  | N  |
|    hd_fetch_reg     | Flip-flop |   1   |  -  | -  | N  | N  | N  | N  | N  |
|     headout_reg     | Flip-flop |  48   |  Y  | N  | N  | N  | N  | N  | N  |
|    pc_state_reg     | Flip-flop |   3   |  Y  | N  | N  | N  | N  | N  | N  |
|     rt_addr_reg     | Flip-flop |  16   |  Y  | N  | N  | N  | N  | N  | N  |
|     rt_read_reg     | Flip-flop |   1   |  -  | -  | N  | N  | N  | N  | N  |
|      rtag_reg       | Flip-flop |  16   |  Y  | N  | N  | N  | N  | N  | N  |
|     rtdata_reg      | Flip-flop |  40   |  Y  | N  | N  | N  | N  | N  | N  |
|     vpivci_reg      | Flip-flop |  24   |  Y  | N  | N  | N  | N  | N  | N  |
===============================================================================

Current design is now 'ht'
1

/* set_operating_conditions wccom */

/* create_clock Clk_AHT_In  -period 25 */
```

```
create_clock Clk_com  -period 20
Performing create_clock on port ’Clk_com’.
1
/* otherwise, the delays would exceed the 25 ns! */

compile -map_effort medium

  Loading target library ’lca300kv’
Warning: Multibit library cell ’FD2X4’ will not be used in multibit optimization. (OPT-918)
Warning: Multibit library cell ’FD2X2’ will not be used in multibit optimization. (OPT-918)
Warning: Multibit library cell ’FD4X4L’ will not be used in multibit optimization. (OPT-918)
Warning: Multibit library cell ’LD1X4’ will not be used in multibit optimization. (OPT-918)
Warning: Multibit library cell ’FD3X4L’ will not be used in multibit optimization. (OPT-918)
Warning: Multibit library cell ’FD3X4’ will not be used in multibit optimization. (OPT-918)
Warning: Multibit library cell ’FD3X2’ will not be used in multibit optimization. (OPT-918)
Warning: Multibit library cell ’FD2X4L’ will not be used in multibit optimization. (OPT-918)
Warning: Multibit library cell ’FD1X4’ will not be used in multibit optimization. (OPT-918)
Warning: Multibit library cell ’FD1X2’ will not be used in multibit optimization. (OPT-918)
Warning: Multibit library cell ’FD4X4’ will not be used in multibit optimization. (OPT-918)
Warning: Multibit library cell ’FD4X2’ will not be used in multibit optimization. (OPT-918)
Warning: Multibit library cell ’LD1X4P’ will not be used in multibit optimization. (OPT-918)
Warning: Multibit library cell ’FD1X4L’ will not be used in multibit optimization. (OPT-918)
Warning: Library cell ’FDN1S’ could not be modeled for sequential mapping. (OPT-1201)
Warning: Library cell ’FDN1SP’ could not be modeled for sequential mapping. (OPT-1201)
  Loading design ’ht’

  Beginning Resource Allocation  (constraint driven)
  -----------------------------
  Structuring ’ht’
  Mapping ’ht’
  Allocating blocks in ’ht’
  Allocating blocks in ’ht’

  Beginning Mapping Optimizations  (Medium effort)
  -------------------------------
  Structuring ’ht’
  Mapping ’ht’
Information: Changed wire load model for ’ht’ from ’B0X0’ to ’B2X2’. (OPT-170)

                                               TOTAL NEG     DESIGN RULE
            TRIALS      AREA     DELTA DELAY     SLACK          COST
           --------    ------    -----------    -------        ------
         0
   --------
         0



  Beginning Phase 1 Design Rule Fixing
  ------------------------------------

                                               TOTAL NEG     DESIGN RULE
            TRIALS      AREA     DELTA DELAY     SLACK          COST
           --------    ------    -----------    -------        ------
         1     3973.1          0.00         0.0           0.7
         1     3972.1          0.00         0.0           0.7
         1     3971.1          0.00         0.0           0.7
         0
   --------
         3



  Beginning Area-Recovery Phase  (cleanup)
  -----------------------------

                                               TOTAL NEG     DESIGN RULE
            TRIALS      AREA     DELTA DELAY     SLACK          COST
           --------    ------    -----------    -------        ------
         4     3956.3          0.00         0.0           0.7
         3     3946.9          0.00         0.0           0.7

trials removed. w.l.

        15     3479.9          0.00         0.0           0.7
         7     3478.9          0.00         0.0           0.7
         0
   --------
      2271


  Optimization Complete
  ---------------------
  Transferring Design ’ht’ to database ’ht.db’
Current design is ’ht’.
1

report_area
Information: Updating design information... (UID-85)

****************************************
Report : area
Design : ht
```

```
Version: 1998.08
Date   : Mon Mar 27 16:33:32 2000
****************************************

Library(s) Used:

    lca300kv (File: /afs/informatik.uni-tuebingen.de/ti/synopsys/1998.8/libraries/syn/lca300kv.db)

Number of ports:              145
Number of nets:               831
Number of cells:              539
Number of references:          20

Combinational area:        402.000000
Noncombinational area:    2134.000000
Net Interconnect area:     942.890564

Total cell area:          2536.000000
Total area:               3478.890625
1

report_timing

****************************************
Report : timing
        -path full
        -delay max
        -max_paths 1
Design : ht
Version: 1998.08
Date   : Mon Mar 27 16:33:32 2000
****************************************

Operating Conditions: NOM   Library: lca300kv
Wire Loading Model Mode: enclosed

Design           Wire Loading Model      Library
------------------------------------------------
ht                     B2X2              lca300kv


  Startpoint: pc_state_reg<2>
              (rising edge-triggered flip-flop clocked by Clk_com)
  Endpoint: Header_reg<4>
            (rising edge-triggered flip-flop clocked by Clk_com)
  Path Group: Clk_com
  Path Type: max

  Point                                    Incr       Path
  -----------------------------------------------------------
  clock Clk_com (rise edge)                0.00       0.00
  clock network delay (ideal)              0.00       0.00
  pc_state_reg<2>/CP (FDS2)                0.00       0.00 r
  pc_state_reg<2>/QN (FDS2)                1.16       1.16 r
  U718/Z (ND4P)                            1.12       2.28 f
  U680/Z (IVP)                             0.28       2.56 r
  U616/Z (ND2)                             1.29       3.85 f
  U605/Z (AO4)                             0.46       4.31 r
  Header_reg<4>/D (FDS2L)                  0.00       4.31 r
  data arrival time                                   4.31

  clock Clk_com (rise edge)               20.00      20.00
  clock network delay (ideal)              0.00      20.00
  Header_reg<4>/CP (FDS2L)                 0.00      20.00 r
  library setup time                      -0.61      19.39
  data required time                                 19.39
  -----------------------------------------------------------
  data required time                                 19.39
  data arrival time                                  -4.31
  -----------------------------------------------------------
  slack (MET)                                        15.08


1

compile -map_effort medium -boundary_optimization

  Loading design ’ht’

  Beginning Mapping Optimizations  (Medium effort)
  -------------------------------
  Structuring ’ht’
  Mapping ’ht’

                                               TOTAL NEG     DESIGN RULE
            TRIALS      AREA     DELTA DELAY     SLACK          COST
           --------    ------    -----------    -------        ------
         0
   --------
         0
  Beginning Phase 1 Design Rule Fixing
  ------------------------------------

                                               TOTAL NEG     DESIGN RULE
            TRIALS      AREA     DELTA DELAY     SLACK          COST
           --------    ------    -----------    -------        ------
         1     4099.9          0.00         0.0           0.7
         1     4098.9          0.00         0.0           0.7
         1     4097.9          0.00         0.0           0.7

trials removed. w.l.

         1     4034.9          0.00         0.0           0.7
         1     4033.9          0.00         0.0           0.7
         0
   --------
        67


  Beginning Area-Recovery Phase  (cleanup)
  -----------------------------

                                               TOTAL NEG     DESIGN RULE
            TRIALS      AREA     DELTA DELAY     SLACK          COST
           --------    ------    -----------    -------        ------
         3     4021.2          0.00         0.0           0.7
         3     4013.5          0.00         0.0           0.7

trials removed. w.l.

        19     3451.5          0.00         0.0           0.7
         7     3450.5          0.00         0.0           0.7
         0
   --------
      2263


  Optimization Complete
  ---------------------
  Transferring Design ’ht’ to database ’ht.db’
Current design is ’ht’.
1

report_area
Information: Updating design information... (UID-85)

****************************************
Report : area
Design : ht
Version: 1998.08
Date   : Mon Mar 27 16:34:14 2000
****************************************

Library(s) Used:

    lca300kv (File: /afs/informatik.uni-tuebingen.de/ti/synopsys/1998.8/libraries/syn/lca300kv.db)

Number of ports:              145
Number of nets:               824
Number of cells:              532
Number of references:          18

Combinational area:        382.000000
Noncombinational area:    2134.000000
Net Interconnect area:     934.453064

Total cell area:          2516.000000
Total area:               3450.453125
1

report_timing

****************************************
Report : timing
        -path full
        -delay max
        -max_paths 1
Design : ht
Version: 1998.08
Date   : Mon Mar 27 16:34:14 2000
****************************************

Operating Conditions: NOM   Library: lca300kv
Wire Loading Model Mode: enclosed

Design           Wire Loading Model      Library
------------------------------------------------
ht                     B2X2              lca300kv


  Startpoint: pc_state_reg<1>
              (rising edge-triggered flip-flop clocked by Clk_com)
  Endpoint: Header_reg<5>
            (rising edge-triggered flip-flop clocked by Clk_com)
  Path Group: Clk_com
  Path Type: max

  Point                                    Incr       Path
  -----------------------------------------------------------
  clock Clk_com (rise edge)                0.00       0.00
  clock network delay (ideal)              0.00       0.00
  pc_state_reg<1>/CP (FDS2)                0.00       0.00 r
  pc_state_reg<1>/QN (FDS2)                1.12       1.12 r
  U1191/Z (ND4)                            1.47       2.60 f
  U1224/Z (IVP)                            0.26       2.86 r
  U1236/Z (ND2)                            0.90       3.75 f
  U1235/Z1 (B2I)                           0.30       4.05 r
  U1235/Z2 (B2I)                           0.44       4.49 f
  U1209/Z (AO4)                            0.28       4.77 r
  Header_reg<5>/D (FDS2L)                  0.00       4.77 r
  data arrival time                                   4.77

  clock Clk_com (rise edge)               20.00      20.00
  clock network delay (ideal)              0.00      20.00
  Header_reg<5>/CP (FDS2L)                 0.00      20.00 r
  library setup time                      -0.59      19.41
  data required time                                 19.41
  -----------------------------------------------------------
  data required time                                 19.41
  data arrival time                                  -4.77
  -----------------------------------------------------------
  slack (MET)                                        14.64


1
quit
1
dc_shell>
Thank you...
```

8.2 Header Translator Module HT

8.2.1 The Dali Synthesis Report for HT

```
Processing /afs/informatik.uni-tuebingen.de/home/wlange/hilan/aht/ht_dali/ht.pdb
Information: Removing prior stored synthesis information from ht.pdb (PCC-9)
Creating initial circuit
Creating action logic for analysis.
Patching control logic...
Optimizing logic structure
style = min_encoded
Analyzing Protocol Including Datapath
shell=0 states=1 (controller states + datapath variables)
shell=1 states=3 (controller states + datapath variables)
shell=2 states=5 (controller states + datapath variables)
optimizing...
shell=3 states=6 (controller states + datapath variables)
shell=4 states=8 (controller states + datapath variables)
shell=5 states=9 (controller states + datapath variables)
shell=6 states=10 (controller states + datapath variables)
shell=7 states=12 (controller states + datapath variables)
shell=8 states=14 (controller states + datapath variables)
states of controller=14 (excludes variables)
Converting to state graph
Pruning unreachable states due to datapath)
Simplifing transitions
Dumping state graph ht.gdl
Launching state graph viewer...
xvcg ht.gdl
Minimizing state graph
Encoding using minimum bits and binary codes
Dumping state graph REDUCED_ht.gdl
Launching state graph viewer...
xvcg REDUCED_ht.gdl
Converting to circuit
Simplifying logic equations.
Using CASE statement action block: 14 actions 8 triggers used, 9 cases

==============================================================================

         Input File:  /afs/informatik.uni-tuebingen.de/home/wlange/hilan/aht/ht_dali/ht.pdb
           Protocol:  ht
      Compile Style:  min_encoded
     Compile Effort:  high

==============================================================================

  Controller Report
                        Initial    Final
                        -------    -----
```

```
             States:        14        8
        Transitions:        25       13
    State Variables:        14        3

     Control Inputs:       288
    Action Triggers:        14

===============================================================================
Cleaning up
Optimizing logic structure
Estimating control logic...
Network optimization of control logic
                      area  delay
                      [ND2] [logic leves]
             initial    118  4.4
               sweep     72  3.8
     structure(area)     57  4.0
      structure(tds)     82  6.0
              redund     60  5.8
               sweep     53  4.8
     structure(area)     53  3.2
               sweep     53  3.2

Estimated size of protocol:
FSM
  combinational area        62.9
  noncombinational area     35.0  (5 bits)
  total area                97.9

Counters required for repeat frames
  combinational area         0.0  (assuming no sharing of decrementers)
  noncombinational area      0.0  (0 bits total)
  total area                 0.0

Variables and ports
  combinational area       424.0  (reset logic)
  noncombinational area   1484.0  (212 bits total)
  total area              1908.0

Combinational logic of
actions is NOT included
                         =======
Total combinational area   486.9
Total noncombinational    1519.0
Total area                2005.9

Delay through the combinational
part of the FSM             2.43 ns (    3.2 logic levels)

Library:
  7.00 gates per 1-bit D flipflop,
  1.00 gates per logical 2-bit NAND operation,
  0.75 ns per logic level
Storing data
Writing ht.pdb+ (synthesized pdb)
```

8.2.2 The Logic Synthesis Report for HT

```
                      Behavioral Compiler (TM)
                        DC Professional (TM)
                           DC Expert (TM)
                         FPGA Compiler (TM)
                         VHDL Compiler (TM)
                          HDL Compiler (TM)
                        Library Compiler (TM)
                         Power Compiler (TM)
                         Test Compiler (TM)
                          ECO Compiler (TM)
                      DesignWare Developer (TM)
                          DesignPower (TM)

                   Version 1998.08 -- Jun 25, 1998
              Copyright (c) 1988-1998 by Synopsys, Inc.
                         ALL RIGHTS RESERVED

This program is proprietary and confidential information of Synopsys, Inc.
and may be used and disclosed only as authorized in a license agreement
controlling such use and disclosure.

Initializing...
/* dc_script fuer rtl-eingangskomponente  - ahtin_rtl.vhd...  */

analyze -f vhdl ht_syn.vhd
Loading db file ’/afs/informatik.uni-tuebingen.de/ti/synopsys/1998.8/libraries/syn/standard.sldb’
Loading db file ’/afs/informatik.uni-tuebingen.de/ti/synopsys/1998.8/libraries/syn/gtech.db’
Loading db file ’/afs/informatik.uni-tuebingen.de/ti/synopsys/1998.8/libraries/syn/lca300kv.db’
Reading in the Synopsys vhdl primitives.
/afs/informatik.uni-tuebingen.de/home/wlange/hilan/aht/ht_dalils/ht_syn.vhd:
1

elaborate ht
```

```
Inferred memory devices in process 'PC_PROC'
in routine ht line 142 in file
         '/afs/informatik.uni-tuebingen.de/home/wlange/hilan/aht/ht_dalils/ht_syn.vhd'.
===============================================================================
|    Register Name    |   Type    | Width | Bus | MB | AR | AS | SR | SS | ST |
===============================================================================
|     Address_reg     | Flip-flop |  16   |  Y  | N  | N  | N  | N  | N  | N  |
|     Header_reg      | Flip-flop |  48   |  N  | N  | ?  | ?  | ?  | ?  | ?  |
|      fwrq_reg       | Flip-flop |   1   |  -  | -  | N  | N  | N  | N  | N  |
|    hd_fetch_reg     | Flip-flop |   1   |  -  | -  | N  | N  | N  | N  | N  |
|     headout_reg     | Flip-flop |  48   |  Y  | N  | N  | N  | N  | N  | N  |
|    pc_state_reg     | Flip-flop |   3   |  Y  | N  | N  | N  | N  | N  | N  |
|     rt_addr_reg     | Flip-flop |  16   |  Y  | N  | N  | N  | N  | N  | N  |
|     rt_read_reg     | Flip-flop |   1   |  -  | -  | N  | N  | N  | N  | N  |
|      rtag_reg       | Flip-flop |  16   |  Y  | N  | N  | N  | N  | N  | N  |
|     rtdata_reg      | Flip-flop |  40   |  Y  | N  | N  | N  | N  | N  | N  |
|     vpivci_reg      | Flip-flop |  24   |  Y  | N  | N  | N  | N  | N  | N  |
===============================================================================

Current design is now 'ht'
1

/* set_operating_conditions wccom */

/* create_clock Clk_AHT_In  -period 25 */

create_clock clk_aht_in  -period 20
Warning: Can't find object 'clk_aht_in' in design 'ht'. (UID-95)
Error: Design object list required for the '<port_pin_list>' argument. (EQN-19)
Usage: create_clock
<port_pin_list>            (list of ports and/or pins)
-name <clock_name>         (name for the clock)
-period <period_value>     (period of the clock)
-waveform <edge_list>      (alternating rise, fall times for 1 period)

0
/* otherwise, the delays would exceed the 25 ns! */

compile -map_effort medium

  Loading target library 'lca300kv'
Warning: Multibit library cell 'FD2X4' will not be used in multibit optimization. (OPT-918)
Warning: Multibit library cell 'FD2X2' will not be used in multibit optimization. (OPT-918)
Warning: Multibit library cell 'FD4X4L' will not be used in multibit optimization. (OPT-918)
Warning: Multibit library cell 'LD1X4' will not be used in multibit optimization. (OPT-918)
Warning: Multibit library cell 'FD3X4L' will not be used in multibit optimization. (OPT-918)
Warning: Multibit library cell 'FD3X4' will not be used in multibit optimization. (OPT-918)
Warning: Multibit library cell 'FD3X2' will not be used in multibit optimization. (OPT-918)
Warning: Multibit library cell 'FD2X4L' will not be used in multibit optimization. (OPT-918)
Warning: Multibit library cell 'FD1X4' will not be used in multibit optimization. (OPT-918)
Warning: Multibit library cell 'FD1X2' will not be used in multibit optimization. (OPT-918)
Warning: Multibit library cell 'FD4X4' will not be used in multibit optimization. (OPT-918)
Warning: Multibit library cell 'FD4X2' will not be used in multibit optimization. (OPT-918)
Warning: Multibit library cell 'LD1X4P' will not be used in multibit optimization. (OPT-918)
Warning: Multibit library cell 'FD1X4L' will not be used in multibit optimization. (OPT-918)
Warning: Library cell 'FDN1S' could not be modeled for sequential mapping. (OPT-1201)
Warning: Library cell 'FDN1SP' could not be modeled for sequential mapping. (OPT-1201)
  Loading design 'ht'
Information: Design 'ht' has no optimization constraints set. (OPT-108)

  Beginning Resource Allocation  (area only)
  -----------------------------
  Allocating blocks in 'ht'
  Allocating blocks in 'ht'

  Beginning Mapping Optimizations  (Medium effort)
  -------------------------------
  Structuring 'ht'
  Mapping 'ht'
Information: Changed wire load model for 'ht' from 'B0X0' to 'B2X2'. (OPT-170)

                                               TOTAL NEG     DESIGN RULE
            TRIALS      AREA     DELTA DELAY     SLACK          COST
           --------    ------    -----------    -------        ------
         0
   --------
         0



  Beginning Phase 1 Design Rule Fixing  (max_capacitance)
  ------------------------------------

                                               TOTAL NEG     DESIGN RULE
            TRIALS      AREA     DELTA DELAY     SLACK          COST
           --------    ------    -----------    -------        ------
         4     3393.2         0.00         0.0          10.6
         4     3393.2         0.00         0.0           9.4
         4     3394.2         0.00         0.0           8.8
         4     3395.2         0.00         0.0           7.7
        11     3396.2         0.00         0.0           7.1
         8     3397.2         0.00         0.0           6.2
         5     3398.2         0.00         0.0           5.2
         9     3399.2          0.00         0.0           5.2
         5     3400.2          0.00         0.0           5.1
         5     3401.2          0.00         0.0           5.1
        73     3402.9          0.00         0.0           3.4
        80     3404.6          0.00         0.0           2.2
        79     3406.3          0.00         0.0           1.5
       105     3408.0          0.00         0.0           0.7
         1     3407.0          0.00         0.0           0.7
         1     3406.0          0.00         0.0           0.7
         1     3405.0          0.00         0.0           0.7
         0
   --------
       399


  Beginning Area-Recovery Phase  (cleanup)
  -----------------------------

                                               TOTAL NEG     DESIGN RULE
            TRIALS      AREA     DELTA DELAY     SLACK          COST
           --------    ------    -----------    -------        ------
         6     3402.3          0.00         0.0           0.7
         2     3399.6          0.00         0.0           0.7
       419     3398.6          0.00         0.0           0.7
       928
   --------
      1355


  Optimization Complete
  ---------------------
  Transferring Design ’ht’ to database ’ht.db’
Current design is ’ht’.
1

report_area
Information: Updating design information... (UID-85)

****************************************
Report : area
Design : ht
Version: 1998.08
Date   : Tue Nov 23 11:05:20 1999
****************************************

Library(s) Used:

    lca300kv (File: /afs/informatik.uni-tuebingen.de/ti/synopsys/1998.8/libraries/syn/lca300kv.db)

Number of ports:              145
Number of nets:               750
Number of cells:              458
Number of references:          21

Combinational area:        381.000000
Noncombinational area:    2138.000000
Net Interconnect area:     879.609314

Total cell area:          2519.000000
Total area:               3398.609375
1

report_timing

****************************************
Report : timing
        -path full
        -delay max
        -max_paths 1
Design : ht
Version: 1998.08
Date   : Tue Nov 23 11:05:20 1999
****************************************

Operating Conditions: NOM   Library: lca300kv
Wire Loading Model Mode: enclosed

Design           Wire Loading Model      Library
------------------------------------------------
ht                     B2X2              lca300kv


  Startpoint: rt_addr_reg<15>
              (rising edge-triggered flip-flop)
  Endpoint: rt_addr<15>
            (output port)
  Path Group: (none)
  Path Type: max

  Point                                    Incr       Path
  -----------------------------------------------------------
  rt_addr_reg<15>/CP (FDS2L)               0.00       0.00 r
  rt_addr_reg<15>/Q (FDS2L)                0.72       0.72 f
  rt_addr<15> (out)                        0.00       0.72 f
  data arrival time                                   0.72
  -----------------------------------------------------------
  (Path is unconstrained)


1

compile -map_effort medium -boundary_optimization

  Loading design ’ht’
Information: Design ’ht’ has no optimization constraints set. (OPT-108)

  Beginning Mapping Optimizations  (Medium effort)
  -------------------------------
  Structuring ’ht’
  Mapping ’ht’

                                               TOTAL NEG     DESIGN RULE
            TRIALS      AREA     DELTA DELAY     SLACK          COST
           --------    ------    -----------    -------        ------
         0
   --------
         0



  Beginning Phase 1 Design Rule Fixing  (max_capacitance)
  ------------------------------------

                                               TOTAL NEG     DESIGN RULE
            TRIALS      AREA     DELTA DELAY     SLACK          COST
           --------    ------    -----------    -------        ------
         4     3583.8          0.00         0.0          10.2

.. trials removed. w.l. ...

         1     3511.9          0.00         0.0           0.7
         1     3510.9          0.00         0.0           0.7
         0
   --------
       383



  Beginning Area-Recovery Phase  (cleanup)
  -----------------------------

                                               TOTAL NEG     DESIGN RULE
            TRIALS      AREA     DELTA DELAY     SLACK          COST
           --------    ------    -----------    -------        ------
         5     3508.2          0.00         0.0           0.7
         2     3505.5          0.00         0.0           0.7

.. trials removed. w.l.

         3     3372.7          0.00         0.0           0.7
         3     3371.7          0.00         0.0           0.7
      1100
   --------
      1484


  Optimization Complete
  ---------------------
  Transferring Design ’ht’ to database ’ht.db’
Current design is ’ht’.
1

report_area
Information: Updating design information... (UID-85)

****************************************
Report : area
Design : ht
Version: 1998.08
Date   : Tue Nov 23 11:05:59 1999
****************************************

Library(s) Used:

    lca300kv (File: /afs/informatik.uni-tuebingen.de/ti/synopsys/1998.8/libraries/syn/lca300kv.db)

Number of ports:              145
Number of nets:               740
Number of cells:              448
Number of references:          22

Combinational area:        359.000000
Noncombinational area:    2138.000000
Net Interconnect area:     874.687439
```

```
Total cell area:          2497.000000
Total area:               3371.687500
1

report_timing

****************************************
Report : timing
        -path full
        -delay max
        -max_paths 1
Design : ht
Version: 1998.08
Date   : Tue Nov 23 11:05:59 1999
****************************************

Operating Conditions: NOM   Library: lca300kv
Wire Loading Model Mode: enclosed

Design           Wire Loading Model      Library
------------------------------------------------
ht                     B2X2              lca300kv


  Startpoint: rt_addr_reg<15>
              (rising edge-triggered flip-flop)
  Endpoint: rt_addr<15>
            (output port)
  Path Group: (none)
  Path Type: max

  Point                                    Incr       Path
  -----------------------------------------------------------
  rt_addr_reg<15>/CP (FDS2L)               0.00       0.00 r
  rt_addr_reg<15>/Q (FDS2L)                0.72       0.72 f
  rt_addr<15> (out)                        0.00       0.72 f
  data arrival time                                   0.72
  -----------------------------------------------------------
  (Path is unconstrained)
```

8.3 Shift Register Control Module SRC

8.3.1 The Dali Synthesis Report for SRC

```
Processing /afs/informatik.uni-tuebingen.de/home/wlange/hilan/aht/src_dali/src.pdb
Information: Removing prior stored synthesis information from src.pdb (PCC-9)
Creating initial circuit
Inserting LFSR for repeat operator count=12
Optimizing logic structure
style = min_encoded
Converting to state graph
Dumping state graph src.gdl
Launching state graph viewer...
xvcg src.gdl
Minimizing state graph
Encoding using minimum bits and binary codes
Dumping state graph REDUCED_src.gdl
Launching state graph viewer...
xvcg REDUCED_src.gdl
Converting to circuit
Simplifying logic equations.
Simplified 25 logic functions
Using CASE statement action block: 38 actions 17 triggers used, 17 cases

==============================================================================

         Input File:  /afs/informatik.uni-tuebingen.de/home/wlange/hilan/aht/src_dali/src.pdb
           Protocol:  src
      Compile Style:  min_encoded
     Compile Effort:  high

==============================================================================

  Controller Report
                        Initial    Final
                        -------    -----

             States:        48       22
        Transitions:        85       36
    State Variables:        26        5

     Control Inputs:       206
    Action Triggers:        38

==============================================================================
Cleaning up
Optimizing logic structure
```

```
Estimating control logic...
Network optimization of control logic
                      area  delay
                      [ND2] [logic leves]
             initial    473  7.6
               sweep    328  6.9
     structure(area)    178  5.8
      structure(tds)    259 10.5
              redund    210  8.3
               sweep    187  7.4
     structure(area)    153  4.4
               sweep    153  4.4

Estimated size of protocol:
FSM
  combinational area       166.9
  noncombinational area     49.0  (7 bits)
  total area               215.9

Counters required for repeat frames
  combinational area        19.0  (assuming no sharing of decrementers)
  noncombinational area     28.0  (4 bits total)
  total area                47.0

Variables and ports
  combinational area       276.0  (reset logic)
  noncombinational area    966.0  (138 bits total)
  total area              1242.0

Combinational logic of
actions is NOT included
                         =======
Total combinational area   461.9
Total noncombinational    1043.0
Total area                1504.9

Delay through the combinational
part of the FSM             3.28 ns (    4.4 logic levels)

Library:
  7.00 gates per 1-bit D flipflop,
  1.00 gates per logical 2-bit NAND operation,
  0.75 ns per logic level
Storing data
Writing src.pdb+ (synthesized pdb)
```

8.3.2 The Logic Synthesis Report for SRC

```
                      Behavioral Compiler (TM)
                        DC Professional (TM)
                           DC Expert (TM)
                         FPGA Compiler (TM)
                         VHDL Compiler (TM)
                          HDL Compiler (TM)
                        Library Compiler (TM)
                         Power Compiler (TM)
                         Test Compiler (TM)
                          ECO Compiler (TM)
                      DesignWare Developer (TM)
                          DesignPower (TM)

                   Version 1998.08 -- Jun 25, 1998
              Copyright (c) 1988-1998 by Synopsys, Inc.
                         ALL RIGHTS RESERVED

This program is proprietary and confidential information of Synopsys, Inc.
and may be used and disclosed only as authorized in a license agreement
controlling such use and disclosure.

Initializing...
/* dc_script fuer rtl-eingangskomponente  - ahtin_rtl.vhd...  */

analyze -f vhdl src/utils_src.vhd
Loading db file ’/afs/informatik.uni-tuebingen.de/ti/synopsys/1998.8/libraries/syn/standard.sldb’
Loading db file ’/afs/informatik.uni-tuebingen.de/ti/synopsys/1998.8/libraries/syn/gtech.db’
Loading db file ’/afs/informatik.uni-tuebingen.de/ti/synopsys/1998.8/libraries/syn/lca300kv.db’
Reading in the Synopsys vhdl primitives.
/afs/informatik.uni-tuebingen.de/home/wlange/hilan/aht/src_dalils/src/utils_src.vhd:
1
analyze -f vhdl src/src_syn.vhd
/afs/informatik.uni-tuebingen.de/home/wlange/hilan/aht/src_dalils/src/src_syn.vhd:
1

elaborate src

Inferred memory devices in process ’PC_PROC’
in routine src line 139 in file
         ’/afs/informatik.uni-tuebingen.de/home/wlange/hilan/aht/src_dalils/src/src_syn.vhd’.
===============================================================================
|    Register Name    |   Type    | Width | Bus | MB | AR | AS | SR | SS | ST |
===============================================================================
|   Cell_loaded_reg   | Flip-flop |   1   |  -  | -  | N  | N  | N  | N  | N  |
|    Cell_read_reg    | Flip-flop |   1   |  -  | -  | N  | N  | N  | N  | N  |
|     aux_reg_reg     | Flip-flop |  32   |  Y  | N  | N  | N  | N  | N  | N  |
|      busy_reg       | Flip-flop |   1   |  -  | -  | N  | N  | N  | N  | N  |
|     load_ok_reg     | Flip-flop |   1   |  -  | -  | N  | N  | N  | N  | N  |
|  pc_counter_0_reg   | Flip-flop |   4   |  Y  | N  | N  | N  | N  | N  | N  |
|    pc_state_reg     | Flip-flop |   6   |  N  | N  | ?  | ?  | ?  | ?  | ?  |
|    shift_reg_reg    | Flip-flop |  48   |  N  | N  | ?  | ?  | ?  | ?  | ?  |
|    shiftreg_reg     | Flip-flop |  48   |  Y  | N  | N  | N  | N  | N  | N  |
|    sr_strobe_reg    | Flip-flop |   1   |  -  | -  | N  | N  | N  | N  | N  |
|    transm_ok_reg    | Flip-flop |   1   |  -  | -  | N  | N  | N  | N  | N  |
===============================================================================

Current design is now 'src'
1

/* set_operating_conditions wccom */

/* create_clock clk_xmit  -period 25 */

create_clock Clk_xmit -period 20
Performing create_clock on port 'Clk_xmit'.
1
/* otherwise, the delays would exceed the 25 ns! */

compile -map_effort medium

  Loading target library 'lca300kv'
Warning: Multibit library cell 'FD2X4' will not be used in multibit optimization. (OPT-918)
Warning: Multibit library cell 'FD2X2' will not be used in multibit optimization. (OPT-918)
Warning: Multibit library cell 'FD4X4L' will not be used in multibit optimization. (OPT-918)
Warning: Multibit library cell 'LD1X4' will not be used in multibit optimization. (OPT-918)
Warning: Multibit library cell 'FD3X4L' will not be used in multibit optimization. (OPT-918)
Warning: Multibit library cell 'FD3X4' will not be used in multibit optimization. (OPT-918)
Warning: Multibit library cell 'FD3X2' will not be used in multibit optimization. (OPT-918)
Warning: Multibit library cell 'FD2X4L' will not be used in multibit optimization. (OPT-918)
Warning: Multibit library cell 'FD1X4' will not be used in multibit optimization. (OPT-918)
Warning: Multibit library cell 'FD1X2' will not be used in multibit optimization. (OPT-918)
Warning: Multibit library cell 'FD4X4' will not be used in multibit optimization. (OPT-918)
Warning: Multibit library cell 'FD4X2' will not be used in multibit optimization. (OPT-918)
Warning: Multibit library cell 'LD1X4P' will not be used in multibit optimization. (OPT-918)
Warning: Multibit library cell 'FD1X4L' will not be used in multibit optimization. (OPT-918)
Warning: Library cell 'FDN1S' could not be modeled for sequential mapping. (OPT-1201)
Warning: Library cell 'FDN1SP' could not be modeled for sequential mapping. (OPT-1201)
  Loading design 'src'

  Beginning Resource Allocation  (constraint driven)
  -----------------------------
  Structuring 'src'
  Mapping 'src'
  Allocating blocks in 'src'
  Allocating blocks in 'src'

  Beginning Mapping Optimizations  (Medium effort)
  -------------------------------
  Structuring 'src'
  Mapping 'src'
Information: Changed wire load model for 'src' from 'B0X0' to 'B2X2'. (OPT-170)

                                              TOTAL NEG     DESIGN RULE
            TRIALS      AREA     DELTA DELAY     SLACK          COST
           --------    ------    -----------    -------        ------
         0
   --------
         0



  Beginning Phase 1 Design Rule Fixing
  -----------------------------------

                                              TOTAL NEG     DESIGN RULE
            TRIALS      AREA     DELTA DELAY     SLACK          COST
           --------    ------    -----------    -------        ------
         1     3962.3          0.00         0.0           0.6
         1     3961.3          0.00         0.0           0.6
         1     3960.3          0.00         0.0           0.6
         0
   --------
         3



  Beginning Area-Recovery Phase  (cleanup)
  -----------------------------

                                              TOTAL NEG     DESIGN RULE
            TRIALS      AREA     DELTA DELAY     SLACK          COST
           --------    ------    -----------    -------        ------
         2     3942.6          0.00         0.0           0.6
         2     3936.9          0.00         0.0           0.6

.. trials removed .. w.l.
```

```
         7     3164.6          0.00         0.0           0.6
         7     3163.6          0.00         0.0           0.6
        13     3162.6          0.00         0.0           0.6
         0
   --------
      3768


  Optimization Complete
  ---------------------
  Transferring Design ’src’ to database ’src.db’
Current design is ’src’.
1

report_area
Information: Updating design information... (UID-85)

****************************************
Report : area
Design : src
Version: 1998.08
Date   : Wed Nov 24 16:28:58 1999
****************************************

Library(s) Used:

    lca300kv (File: /afs/informatik.uni-tuebingen.de/ti/synopsys/1998.8/libraries/syn/lca300kv.db)

Number of ports:              124
Number of nets:               921
Number of cells:              667
Number of references:          22

Combinational area:        647.000000
Noncombinational area:    1437.000000
Net Interconnect area:    1078.593628

Total cell area:          2084.000000
Total area:               3162.593750
1

report_timing

****************************************
Report : timing
        -path full
        -delay max
        -max_paths 1
Design : src
Version: 1998.08
Date   : Wed Nov 24 16:28:58 1999
****************************************

Operating Conditions: NOM   Library: lca300kv
Wire Loading Model Mode: enclosed

Design           Wire Loading Model      Library
------------------------------------------------
src                    B2X2              lca300kv


  Startpoint: pc_state_reg<1>
              (rising edge-triggered flip-flop clocked by Clk_xmit)
  Endpoint: aux_reg_reg<0>
            (rising edge-triggered flip-flop clocked by Clk_xmit)
  Path Group: Clk_xmit
  Path Type: max

  Point                                    Incr       Path
  -----------------------------------------------------------
  clock Clk_xmit (rise edge)               0.00       0.00
  clock network delay (ideal)              0.00       0.00
  pc_state_reg<1>/CP (FD1)                 0.00       0.00 r
  pc_state_reg<1>/QN (FD1)                 1.03       1.03 f
  U2138/Z (ND2)                            0.55       1.58 r
  U2580/Z (OR2P)                           0.47       2.05 r
  U2450/Z (IVP)                            0.22       2.27 f
  U2389/Z (AO7)                            0.75       3.02 r
  U2360/Z (ND2)                            1.21       4.23 f
  U2392/Z (IVP)                            0.31       4.54 r
  U2562/Z (IVP)                            0.40       4.94 f
  U2390/Z (AO4)                            0.38       5.32 r
  aux_reg_reg<0>/D (FDS2L)                 0.00       5.32 r
  data arrival time                                   5.32

  clock Clk_xmit (rise edge)              20.00      20.00
  clock network delay (ideal)              0.00      20.00
  aux_reg_reg<0>/CP (FDS2L)                0.00      20.00 r
  library setup time                      -0.58      19.42
  data required time                                 19.42
  -----------------------------------------------------------
  data required time                                 19.42
  data arrival time                                  -5.32
  -----------------------------------------------------------
  slack (MET)                                        14.10


1

compile -map_effort medium -boundary_optimization

  Loading design 'src'

  Beginning Mapping Optimizations  (Medium effort)
  -------------------------------
  Structuring 'src'
  Mapping 'src'

                                               TOTAL NEG     DESIGN RULE
            TRIALS      AREA     DELTA DELAY     SLACK          COST
           --------    ------    -----------    -------        ------
         0
   --------
         0



  Beginning Phase 1 Design Rule Fixing
  ------------------------------------

                                               TOTAL NEG     DESIGN RULE
            TRIALS      AREA     DELTA DELAY     SLACK          COST
           --------    ------    -----------    -------        ------
         1     3783.1          0.00         0.0           0.6

.. trials removed .. w.l.

         1     3734.1          0.00         0.0           0.6
         1     3733.1          0.00         0.0           0.6
         0
   --------
        51



  Beginning Area-Recovery Phase  (cleanup)
  -----------------------------

                                               TOTAL NEG     DESIGN RULE
            TRIALS      AREA     DELTA DELAY     SLACK          COST
           --------    ------    -----------    -------        ------
         3     3725.4          0.00         0.0           0.6
.. trials removed.. w.l.
         1     2977.4          0.00         0.0           0.6
         0
   --------
      3392


  Optimization Complete
  ---------------------
  Transferring Design 'src' to database 'src.db'
Current design is 'src'.
1

report_area
Information: Updating design information... (UID-85)

****************************************
Report : area
Design : src
Version: 1998.08
Date   : Wed Nov 24 16:29:54 1999
****************************************

Library(s) Used:

    lca300kv (File: /afs/informatik.uni-tuebingen.de/ti/synopsys/1998.8/libraries/syn/lca300kv.db)

Number of ports:              124
Number of nets:               816
Number of cells:              561
Number of references:          22

Combinational area:        582.000000
Noncombinational area:    1437.000000
Net Interconnect area:     958.359314

Total cell area:          2019.000000
Total area:               2977.359375
1

report_timing

****************************************
Report : timing
        -path full
        -delay max
        -max_paths 1
Design : src
Version: 1998.08
Date   : Wed Nov 24 16:29:54 1999
****************************************

Operating Conditions: NOM   Library: lca300kv
Wire Loading Model Mode: enclosed

Design           Wire Loading Model      Library
------------------------------------------------
src                    B2X2              lca300kv


  Startpoint: pc_state_reg<3>
              (rising edge-triggered flip-flop clocked by Clk_xmit)
  Endpoint: aux_reg_reg<1>
            (rising edge-triggered flip-flop clocked by Clk_xmit)
  Path Group: Clk_xmit
  Path Type: max

  Point                                    Incr       Path
  -----------------------------------------------------------
  clock Clk_xmit (rise edge)               0.00       0.00
  clock network delay (ideal)              0.00       0.00
  pc_state_reg<3>/CP (FD1)                 0.00       0.00 r
  pc_state_reg<3>/Q (FD1)                  0.79       0.79 f
  U3115/Z (IVP)                            0.07       0.86 r
  U2778/Y (IVDA)                           0.57       1.43 f
  U2778/Z (IVDA)                           0.49       1.92 r
  U2826/Z (ND2)                            0.56       2.48 f
  U2756/Z (NR2)                            0.26       2.74 r
  U2754/Z (AO7)                            0.40       3.14 f
  U3109/Z (ND2)                            0.77       3.91 r
  U3108/Z1 (B2I)                           0.37       4.28 f
  U3108/Z2 (B2I)                           0.59       4.87 r
  U3124/Z (OR2P)                           0.78       5.64 r
  U3107/Z1 (B2I)                           0.30       5.95 f
  U3107/Z2 (B2I)                           0.49       6.44 r
  aux_reg_reg<1>/LD (FDS2L)                0.00       6.44 r
  data arrival time                                   6.44

  clock Clk_xmit (rise edge)              20.00      20.00
  clock network delay (ideal)              0.00      20.00
  aux_reg_reg<1>/CP (FDS2L)                0.00      20.00 r
  library setup time                      -0.76      19.24
  data required time                                 19.24
  -----------------------------------------------------------
  data required time                                 19.24
  data arrival time                                  -6.44
  -----------------------------------------------------------
  slack (MET)                                        12.80


1
```

8.4 Receive Data Module RD

8.4.1 The Dali Synthesis Report for RD

```
Processing /afs/informatik.uni-tuebingen.de/home/wlange/hilan/aht/rd_dali/rd.pdb
Creating initial circuit
Inserting LFSR for repeat operator count=12
Creating action logic for analysis.
Patching control logic...
Optimizing logic structure
style = automatic
Analyzing Protocol Including Datapath

.. optimizing steps removed .. WL.

==============================================================================

         Input File:  /afs/informatik.uni-tuebingen.de/home/wlange/hilan/aht/rd_dali/rd.pdb
           Protocol:  rd
      Compile Style:  automatic
     Compile Effort:  high

==============================================================================

  Controller Report
                        Initial    Final
                        -------    -----

             States:       261    (18,5)
        Transitions:        --    (38,13)
```

```
    State Variables:        28    9=(5,3)+1

     Control Inputs:       245
    Action Triggers:        43

==============================================================================
Cleaning up
Optimizing logic structure
Estimating control logic...
Network optimization of control logic
                      area  delay
                      [ND2] [logic leves]
             initial    503  7.2
               sweep    295  6.3
     structure(area)    185  4.9
      structure(tds)    241  7.4
              redund    178  6.3
               sweep    171  6.3
     structure(area)    165  4.7
               sweep    165  4.7

Estimated size of protocol:
FSM
  combinational area       184.6
  noncombinational area     70.0  (10 bits)
  total area               254.6

Counters required for repeat frames
  combinational area        19.0  (assuming no sharing of decrementers)
  noncombinational area     28.0  (4 bits total)
  total area                47.0

Variables and ports
  combinational area       388.0  (reset logic)
  noncombinational area   1358.0  (194 bits total)
  total area              1746.0

Combinational logic of
actions is NOT included
                         =======
Total combinational area   591.6
Total noncombinational    1456.0
Total area                2047.6

Delay through the combinational
part of the FSM             3.53 ns (    4.7 logic levels)

Library:
  7.00 gates per 1-bit D flipflop,
  1.00 gates per logical 2-bit NAND operation,
  0.75 ns per logic level
Storing data
Writing rd.pdb+ (synthesized pdb)
```

8.4.2 The Logic Synthesis Report for RD

```
                      Behavioral Compiler (TM)
                        DC Professional (TM)
                           DC Expert (TM)
                         FPGA Compiler (TM)
                         VHDL Compiler (TM)
                          HDL Compiler (TM)
                        Library Compiler (TM)
                         Power Compiler (TM)
                         Test Compiler (TM)
                          ECO Compiler (TM)
                      DesignWare Developer (TM)
                          DesignPower (TM)

                   Version 1998.08 -- Jun 25, 1998
              Copyright (c) 1988-1998 by Synopsys, Inc.
                         ALL RIGHTS RESERVED

This program is proprietary and confidential information of Synopsys, Inc.
and may be used and disclosed only as authorized in a license agreement
controlling such use and disclosure.

Initializing...
/* dc_script fuer rtl-eingangskomponente  - ahtin_rtl.vhd...  */

analyze -f vhdl src/utilsrd.vhd
Loading db file ’/afs/informatik.uni-tuebingen.de/ti/synopsys/1998.8/libraries/syn/standard.sldb’
Loading db file ’/afs/informatik.uni-tuebingen.de/ti/synopsys/1998.8/libraries/syn/gtech.db’
Loading db file ’/afs/informatik.uni-tuebingen.de/ti/synopsys/1998.8/libraries/syn/lca300kv.db’
Reading in the Synopsys vhdl primitives.
/afs/informatik.uni-tuebingen.de/home/wlange/hilan/aht/rd_dalils/src/utilsrd.vhd:
1
analyze -f vhdl src/rd_syn.vhd
/afs/informatik.uni-tuebingen.de/home/wlange/hilan/aht/rd_dalils/src/rd_syn.vhd:
1
```

```
elaborate rd

Inferred memory devices in process 'PC_PROC_SEQ'
in routine rd line 275 in
         file
         '/afs/informatik.uni-tuebingen.de/home/wlange/hilan/aht/rd_dalils/src/rd_syn.vhd'.
===============================================================================
|    Register Name    |   Type    | Width | Bus | MB | AR | AS | SR | SS | ST |
===============================================================================
|       err_reg       | Flip-flop |   5   |  Y  | N  | N  | N  | N  | N  | N  |
|    hilf_reg_reg     | Flip-flop |  48   |  Y  | N  | N  | N  | N  | N  | N  |
|    out_reg_h_reg    | Flip-flop |  32   |  Y  | N  | N  | N  | N  | N  | N  |
|     out_reg_reg     | Flip-flop |  32   |  Y  | N  | N  | N  | N  | N  | N  |
|  pc_counter_0_reg   | Flip-flop |   4   |  Y  | N  | N  | N  | N  | N  | N  |
|  pc_reg_error_reg   | Flip-flop |   5   |  Y  | N  | N  | N  | N  | N  | N  |
| pc_reg_rcell_bus_reg | Flip-flop |  32   |  Y  | N  | N  | N  | N  | N  | N  |
| pc_reg_rcell_rdy_reg | Flip-flop |  1   |  -  | -  | N  | N  | N  | N  | N  |
| pc_reg_srdta_taken_reg | Flip-flop | 1   |  -  | -  | N  | N  | N  | N  | N  |
|     wr_ack_reg      | Flip-flop |   1   |  -  | -  | N  | N  | N  | N  | N  |
|   write_fifo_reg    | Flip-flop |   1   |  -  | -  | N  | N  | N  | N  | N  |
===============================================================================


Inferred memory devices in process 'PC_PROC_STATE'
in routine rd line 325 in
         file
         '/afs/informatik.uni-tuebingen.de/home/wlange/hilan/aht/rd_dalils/src/rd_syn.vhd'.
===============================================================================
|    Register Name    |   Type    | Width | Bus | MB | AR | AS | SR | SS | ST |
===============================================================================
|    pc_state_reg     | Flip-flop |   9   |  N  | N  | ?  | ?  | ?  | ?  | ?  |
===============================================================================

Current design is now 'rd'
1

/* set_operating_conditions wccom */

/* create_clock clk_com  -period 25 */

create_clock Clk_com -period 20
Performing create_clock on port 'Clk_com'.
1
/* otherwise, the delays would exceed the 25 ns! */

compile -map_effort medium

  Loading target library 'lca300kv'
Warning: Multibit library cell 'FD2X4' will not be used in multibit optimization. (OPT-918)
Warning: Multibit library cell 'FD2X2' will not be used in multibit optimization. (OPT-918)
Warning: Multibit library cell 'FD4X4L' will not be used in multibit optimization. (OPT-918)
Warning: Multibit library cell 'LD1X4' will not be used in multibit optimization. (OPT-918)
Warning: Multibit library cell 'FD3X4L' will not be used in multibit optimization. (OPT-918)
Warning: Multibit library cell 'FD3X4' will not be used in multibit optimization. (OPT-918)
Warning: Multibit library cell 'FD3X2' will not be used in multibit optimization. (OPT-918)
Warning: Multibit library cell 'FD2X4L' will not be used in multibit optimization. (OPT-918)
Warning: Multibit library cell 'FD1X4' will not be used in multibit optimization. (OPT-918)
Warning: Multibit library cell 'FD1X2' will not be used in multibit optimization. (OPT-918)
Warning: Multibit library cell 'FD4X4' will not be used in multibit optimization. (OPT-918)
Warning: Multibit library cell 'FD4X2' will not be used in multibit optimization. (OPT-918)
Warning: Multibit library cell 'LD1X4P' will not be used in multibit optimization. (OPT-918)
Warning: Multibit library cell 'FD1X4L' will not be used in multibit optimization. (OPT-918)
Warning: Library cell 'FDN1S' could not be modeled for sequential mapping. (OPT-1201)
Warning: Library cell 'FDN1SP' could not be modeled for sequential mapping. (OPT-1201)
  Loading design 'rd'

  Beginning Resource Allocation  (constraint driven)
  -----------------------------
  Structuring 'rd'
  Mapping 'rd'
  Allocating blocks in 'rd'
  Allocating blocks in 'rd'

  Beginning Mapping Optimizations  (Medium effort)
  -------------------------------
  Structuring 'rd'
  Mapping 'rd'
Information: Changed wire load model for 'rd' from 'B0X0' to 'B3X3'. (OPT-170)

                                               TOTAL NEG     DESIGN RULE
            TRIALS      AREA     DELTA DELAY     SLACK          COST
           --------    ------    -----------    -------        ------
         0
   --------
         0



  Beginning Phase 1 Design Rule Fixing
  ------------------------------------

                                               TOTAL NEG     DESIGN RULE
            TRIALS      AREA     DELTA DELAY     SLACK          COST
           --------    ------    -----------    -------        ------
         0
   --------
         0



  Beginning Area-Recovery Phase  (cleanup)
  -----------------------------

                                               TOTAL NEG     DESIGN RULE
            TRIALS      AREA     DELTA DELAY     SLACK          COST
           --------    ------    -----------    -------        ------
         4     5474.4          0.00         0.0           0.3
         3     5464.8          0.00         0.0           0.3

.. trials removed.. wl.

        21     4031.4          0.00         0.0           0.3
        15     4030.4          0.00         0.0           0.3
Information: Changed wire load model for ’rd’ from ’B3X3’ to ’B2X2’. (OPT-170)
         0
   --------
      8554


  Optimization Complete
  ---------------------
  Transferring Design ’rd’ to database ’rd.db’
Current design is ’rd’.
1

report_area
Information: Updating design information... (UID-85)

****************************************
Report : area
Design : rd
Version: 1998.08
Date   : Wed Nov 24 17:34:00 1999
****************************************

Library(s) Used:

    lca300kv (File: /afs/informatik.uni-tuebingen.de/ti/synopsys/1998.8/libraries/syn/lca300kv.db)

Number of ports:               92
Number of nets:              1103
Number of cells:              944
Number of references:          21

Combinational area:       1108.000000
Noncombinational area:    1183.000000
Net Interconnect area:    1552.499878

Total cell area:          2291.000000
Total area:               3843.500000
1

report_timing

****************************************
Report : timing
        -path full
        -delay max
        -max_paths 1
Design : rd
Version: 1998.08
Date   : Wed Nov 24 17:34:01 1999
****************************************

Operating Conditions: NOM   Library: lca300kv
Wire Loading Model Mode: enclosed

Design           Wire Loading Model      Library
------------------------------------------------
rd                     B2X2              lca300kv


  Startpoint: pc_state_reg<3>
              (rising edge-triggered flip-flop clocked by Clk_com)
  Endpoint: out_reg_h_reg<0>
            (rising edge-triggered flip-flop clocked by Clk_com)
  Path Group: Clk_com
  Path Type: max

  Point                                    Incr       Path
  -----------------------------------------------------------
  clock Clk_com (rise edge)                0.00       0.00
  clock network delay (ideal)              0.00       0.00
  pc_state_reg<3>/CP (FD1)                 0.00       0.00 r
  pc_state_reg<3>/Q (FD1)                  0.75       0.75 r
  U3025/Y (IVDA)                           0.17       0.92 f
  U3025/Z (IVDA)                           0.57       1.49 r
  U3024/Z (ND2)                            1.01       2.50 f
  U2345/Z (NR2)                            0.33       2.82 r
  U2523/Z (ND2)                            0.41       3.23 f
  U2522/Z (ND2)                            0.22       3.45 r
  U2739/Z (ND2)                            1.24       4.69 f
  U2818/Z (IVP)                            0.32       5.01 r
  U3009/Z (IVP)                            0.38       5.40 f
  U2816/Z (AO7)                            0.26       5.65 r
  out_reg_h_reg<0>/D (FD1)                 0.00       5.65 r
  data arrival time                                   5.65

  clock Clk_com (rise edge)               20.00      20.00
  clock network delay (ideal)              0.00      20.00
  out_reg_h_reg<0>/CP (FD1)                0.00      20.00 r
  library setup time                      -0.20      19.80
  data required time                                 19.80
  -----------------------------------------------------------
  data required time                                 19.80
  data arrival time                                  -5.65
  -----------------------------------------------------------
  slack (MET)                                        14.15


1

compile -map_effort medium -boundary_optimization

  Loading design ’rd’
Warning: In design ’rd’, there are 3 ports not connected to any nets. (LINT-30)

Information: Use the ’check_design’ command for
 more information about warnings. (LINT-99)


  Beginning Mapping Optimizations  (Medium effort)
  -------------------------------
Information: Changed wire load model for ’rd’ from ’B2X2’ to ’B3X3’. (OPT-170)
  Structuring ’rd’
  Mapping ’rd’

                                               TOTAL NEG     DESIGN RULE
            TRIALS      AREA     DELTA DELAY     SLACK          COST
           --------    ------    -----------    -------        ------
         0
   --------
         0



  Beginning Phase 1 Design Rule Fixing
  ------------------------------------

                                               TOTAL NEG     DESIGN RULE
            TRIALS      AREA     DELTA DELAY     SLACK          COST
           --------    ------    -----------    -------        ------
         1     5186.5          0.00         0.0           0.3
         1     5185.5          0.00         0.0           0.3

.. trials removed.. w.l.

         1     5151.5          0.00         0.0           0.3
         1     5150.5          0.00         0.0           0.3
         0
   --------
        37



  Beginning Area-Recovery Phase  (cleanup)
  -----------------------------

                                               TOTAL NEG     DESIGN RULE
            TRIALS      AREA     DELTA DELAY     SLACK          COST
           --------    ------    -----------    -------        ------
         3     5139.7          0.00         0.0           0.3
         2     5131.9          0.00         0.0           0.3
         2     5124.1          0.00         0.0           0.3

.. trials removed.. w.l.

         9     3742.0          0.00         0.0           0.3
        19     3741.0          0.00         0.0           0.3
Information: Changed wire load model for ’rd’ from ’B3X3’ to ’B2X2’. (OPT-170)
         0
   --------
      6283


  Optimization Complete
  ---------------------
  Transferring Design ’rd’ to database ’rd.db’
Current design is ’rd’.
1
```

```
report_area
Information: Updating design information... (UID-85)

****************************************
Report : area
Design : rd
Version: 1998.08
Date   : Wed Nov 24 17:35:22 1999
****************************************

Library(s) Used:

    lca300kv (File: /afs/informatik.uni-tuebingen.de/ti/synopsys/1998.8/libraries/syn/lca300kv.db)

Number of ports:               92
Number of nets:               997
Number of cells:              825
Number of references:          24

Combinational area:       1003.000000
Noncombinational area:    1183.000000
Net Interconnect area:    1387.968628

Total cell area:          2186.000000
Total area:               3573.968750
1

report_timing

****************************************
Report : timing
        -path full
        -delay max
        -max_paths 1
Design : rd
Version: 1998.08
Date   : Wed Nov 24 17:35:22 1999
****************************************

Operating Conditions: NOM   Library: lca300kv
Wire Loading Model Mode: enclosed

Design           Wire Loading Model      Library
------------------------------------------------
rd                     B2X2              lca300kv


  Startpoint: pc_state_reg<2>
              (rising edge-triggered flip-flop clocked by Clk_com)
  Endpoint: out_reg_reg<0>
            (rising edge-triggered flip-flop clocked by Clk_com)
  Path Group: Clk_com
  Path Type: max

  Point                                    Incr       Path
  -----------------------------------------------------------
  clock Clk_com (rise edge)                0.00       0.00
  clock network delay (ideal)              0.00       0.00
  pc_state_reg<2>/CP (FD1)                 0.00       0.00 r
  pc_state_reg<2>/Q (FD1)                  0.73       0.73 r
  U3662/Z (IVP)                            0.10       0.82 f
  U3661/Y (IVDA)                           0.64       1.47 r
  U3656/Z (ND2)                            0.96       2.42 f
  U3632/Z (IVP)                            0.26       2.68 r
  U3114/Z (ND2)                            0.37       3.05 f
  U3695/Z (AN2P)                           0.27       3.32 f
  U3268/Z (AO6)                            0.28       3.60 r
  U3663/Z (OR2P)                           0.37       3.97 r
  U3676/Z (OR2P)                           0.27       4.24 r
  U3639/Z (IVP)                            0.18       4.42 f
  U3316/Z (ND2)                            0.98       5.40 r
  U3644/Z1 (B2I)                           0.43       5.83 f
  U3644/Z2 (B2I)                           0.49       6.31 r
  U3371/Z (AO7)                            0.36       6.67 f
  out_reg_reg<0>/D (FD1)                   0.00       6.67 f
  data arrival time                                   6.67

  clock Clk_com (rise edge)               20.00      20.00
  clock network delay (ideal)              0.00      20.00
  out_reg_reg<0>/CP (FD1)                  0.00      20.00 r
  library setup time                      -0.13      19.87
  data required time                                 19.87
  -----------------------------------------------------------
  data required time                                 19.87
  data arrival time                                  -6.67
  -----------------------------------------------------------
  slack (MET)                                        13.19


1
```

8.5 The AHT Output Module AHT_Out

8.5.1 The Dali Synthesis Report for AHT_Out

```
Processing /afs/informatik.uni-tuebingen.de/home/wlange/hilan/aht/ahtout_dali/ahtout.pdb
Creating initial circuit
Inserting LFSR for repeat operator count=11
Inserting LFSR for repeat operator count=11
Creating action logic for analysis.
Patching control logic...
Optimizing logic structure
style = automatic
Analyzing Protocol Including Datapath
shell=0 states=1 (controller states + datapath variables)
shell=1 states=4 (controller states + datapath variables)
shell=2 states=10 (controller states + datapath variables)
optimizing...

 .. optimizing shells 3 to 24 removed .. wl

states of controller=543 (excludes variables)---------------------------------------------------------------------------

: node :   states  :  state vars  : name                              :
---------------------------------------------------------------------------
:    0 :       543 : [   1:30  ]  : AHTOUT
:    1 :       543 : [   1:30  ]  : alternative
:    2 :        23 : [   1:13  ]  : repeat+
:    3 :        23 : [   1:13  ]  : alternative
:    4 :        22 : [   1:12  ]  : if
:    5 :        22 : [   1:12  ]  : fetch_cell
:    6 :        22 : [   1:12  ]  : fetch_cell
:    7 :        22 : [   1:12  ]  : sequence
:    9 :         3 : [   2:3   ]  : Cont_On
:   11 :         3 : [   2:3   ]  : Cont_On
:   12 :         3 : [   2:3   ]  : sequence
:   18 :         3 : [   4:5   ]  : Cont_On
:   20 :         3 : [   4:5   ]  : Cont_On
:   21 :         3 : [   4:5   ]  : sequence
:   27 :        10 : [   6:10  ]  : repeat
:   28 :         5 : [   6:9   ]  : sequence
:   29 :         3 : [   6:7   ]  : Cont_On
:   31 :         3 : [   6:7   ]  : Cont_On
:   32 :         3 : [   6:7   ]  : sequence
:   38 :         3 : [   8:9   ]  : Cont_On
:   40 :         3 : [   8:9   ]  : Cont_On
:   41 :         3 : [   8:9   ]  : sequence
:   47 :         3 : [  11:12  ]  : Cont_On
:   49 :         3 : [  11:12  ]  : Cont_On
:   50 :         3 : [  11:12  ]  : sequence
:   58 :        29 : [  14:30  ]  : repeat+
:   59 :        29 : [  14:30  ]  : alternative
:   60 :        28 : [  14:29  ]  : if
:   61 :        28 : [  14:29  ]  : xmit_cell
:   62 :        28 : [  14:29  ]  : xmit_cell
:   63 :        28 : [  14:29  ]  : sequence
:   64 :         3 : [  14:15  ]  : Cont_On
:   66 :         3 : [  14:15  ]  : Cont_On
:   67 :         3 : [  14:15  ]  : sequence
:   78 :        10 : [  21:25  ]  : repeat
:   79 :         5 : [  21:24  ]  : xmit_4bytes
:   80 :         5 : [  21:24  ]  : xmit_4bytes
:   81 :         5 : [  21:24  ]  : sequence
---------------------------------------------------------------------------
Partitioning node    2 : states=  23 : Frame AHTOUT, repeat ’+’. <ahtout-AHTOUT-3>
Partitioning node   58 : states=  29 : Frame AHTOUT, repeat ’+’. <ahtout-AHTOUT-9>
Compiling partitions
-------------------------------------------------------------------------------
Optimizing /2
-------------------------------------------------------------------------------
Extracting ahtout_part_/2
Initial (13 state vars, 15 actions)
23 states
Creating state graph for ahtout_part_/2
Simplifing transitions
Simplified 40 transition(s)
Dumping state graph ahtout_part_.2.gdl
Initial (23 states, 49 transitions)
Minimizing state graph ahtout_part_/2
Final (10 states, 23 transitions)
Encoding using minimum bits and binary codes
Dumping state graph REDUCED_ahtout_part_.2.gdl
Launching state graph viewer...
xvcg REDUCED_ahtout_part_.2.gdl
Converting to circuit REDUCED_ahtout_part_/2
Simplifying logic equations.
Simplified 15 logic functions
Using CASE statement action block: 15 actions 8 triggers used, 7 cases
Merging VC_MINENC_REDUCED_ahtout_part_/2
Final (4 state vars, 16 actions)
-------------------------------------------------------------------------------
Optimizing /58
```

```
------------------------------------------------------------------------------
Extracting ahtout_part_/58
Initial (17 state vars, 27 actions)
29 states
Creating state graph for ahtout_part_/58
Simplifing transitions
Simplified 28 transition(s)
Dumping state graph ahtout_part_.58.gdl
Initial (29 states, 41 transitions)
Minimizing state graph ahtout_part_/58
Final (20 states, 27 transitions)
Encoding using minimum bits and binary codes
Dumping state graph REDUCED_ahtout_part_.58.gdl
Launching state graph viewer...
xvcg REDUCED_ahtout_part_.58.gdl
Converting to circuit REDUCED_ahtout_part_/58
Simplifying logic equations.
Simplified 22 logic functions
Using CASE statement action block: 27 actions 16 triggers used, 15 cases
Merging VC_MINENC_REDUCED_ahtout_part_/58
Final (5 state vars, 28 actions)
-----------------------------------------------------------------------------
Final wiring of ahtout_part
-----------------------------------------------------------------------------
Top level controller implemented in distributed style using 1 state var
-----------------------------------------------------------------------------

==============================================================================

         Input File:  /afs/informatik.uni-tuebingen.de/home/wlange/hilan/aht/ahtout_dali/ahtout.pdb
           Protocol:  ahtout
      Compile Style:  automatic
     Compile Effort:  high

==============================================================================

  Controller Report
                        Initial    Final
                        -------    -----

             States:       543    (10,20)
        Transitions:        --    (23,27)
    State Variables:        31    10=(4,5)+1

     Control Inputs:       120
    Action Triggers:        42

==============================================================================
Cleaning up
Optimizing logic structure
Estimating control logic...
Network optimization of control logic
                      area  delay
                      [ND2] [logic leves]
             initial    538  6.1
               sweep    328  5.3
     structure(area)    185  4.2
      structure(tds)    266  7.9
              redund    204  6.0
               sweep    193  5.5
     structure(area)    171  3.5
               sweep    171  3.5

Estimated size of protocol:
FSM
  combinational area       192.5
  noncombinational area     77.0  (11 bits)
  total area               269.5

Counters required for repeat frames
  combinational area        38.0  (assuming no sharing of decrementers)
  noncombinational area     56.0  (8 bits total)
  total area                94.0

Variables and ports
  combinational area       172.0  (reset logic)
  noncombinational area    602.0  (86 bits total)
  total area               774.0

Combinational logic of
actions is NOT included
                         =======
Total combinational area   402.5
Total noncombinational     735.0
Total area                1137.5

Delay through the combinational
part of the FSM             2.61 ns (    3.5 logic levels)

Library:
  7.00 gates per 1-bit D flipflop,
  1.00 gates per logical 2-bit NAND operation,
  0.75 ns per logic level
Storing data
```

```
Writing ahtout.pdb+ (synthesized pdb)
```

8.5.2 The Logic Synthesis Report for AHT_Out

```
                      Behavioral Compiler (TM)
                        DC Professional (TM)
                           DC Expert (TM)
                         FPGA Compiler (TM)
                         VHDL Compiler (TM)
                          HDL Compiler (TM)
                        Library Compiler (TM)
                         Power Compiler (TM)
                         Test Compiler (TM)
                          ECO Compiler (TM)
                      DesignWare Developer (TM)
                          DesignPower (TM)

                   Version 1998.08 -- Jun 25, 1998
              Copyright (c) 1988-1998 by Synopsys, Inc.
                         ALL RIGHTS RESERVED

This program is proprietary and confidential information of Synopsys, Inc.
and may be used and disclosed only as authorized in a license agreement
controlling such use and disclosure.

Initializing...
/* dc_script fuer rtl-eingangskomponente  - ahtin_rtl.vhd...  */

analyze -f vhdl ahtout_syn.vhd
Loading db file '/afs/informatik.uni-tuebingen.de/ti/synopsys/1998.8/libraries/syn/standard.sldb'
Loading db file '/afs/informatik.uni-tuebingen.de/ti/synopsys/1998.8/libraries/syn/gtech.db'
Loading db file '/afs/informatik.uni-tuebingen.de/ti/synopsys/1998.8/libraries/syn/lca300kv.db'
Reading in the Synopsys vhdl primitives.
/afs/informatik.uni-tuebingen.de/home/wlange/hilan/aht/ahtout_dalils/ahtout_syn.vhd:
1

elaborate ahtout

Inferred memory devices in process 'PC_PROC_SEQ'
in routine ahtout line 280
         in file
         '/afs/informatik.uni-tuebingen.de/home/wlange/hilan/aht/ahtout_dalils/ahtout_syn.vhd'.
===============================================================================
|    Register Name    |   Type    | Width | Bus | MB | AR | AS | SR | SS | ST |
===============================================================================
|    cell_reg_reg     | Flip-flop |  32   |  Y  | N  | N  | N  | N  | N  | N  |
|      hilf_reg       | Flip-flop |  32   |  Y  | N  | N  | N  | N  | N  | N  |
|  pc_counter_0_reg   | Flip-flop |   4   |  Y  | N  | N  | N  | N  | N  | N  |
|  pc_counter_1_reg   | Flip-flop |   4   |  Y  | N  | N  | N  | N  | N  | N  |
| pc_reg_cell_presync_reg | Flip-flop | 1 |  -  | -  | N  | N  | N  | N  | N  |
| pc_reg_cell_sync_out_reg | Flip-flop | 1 | -  | -  | N  | N  | N  | N  | N  |
| pc_reg_data_out_reg | Flip-flop |   8   |  Y  | N  | N  | N  | N  | N  | N  |
| pc_reg_rcell_read_reg | Flip-flop |  1  |  -  | -  | N  | N  | N  | N  | N  |
| pc_reg_rnew_cell_reg | Flip-flop |  1   |  -  | -  | N  | N  | N  | N  | N  |
|    reg_taken_reg    | Flip-flop |   1   |  -  | -  | N  | N  | N  | N  | N  |
|    transmit_reg     | Flip-flop |   1   |  -  | -  | N  | N  | N  | N  | N  |
===============================================================================


Inferred memory devices in process 'PC_PROC_STATE'
in routine ahtout line 330
         in file
         '/afs/informatik.uni-tuebingen.de/home/wlange/hilan/aht/ahtout_dalils/ahtout_syn.vhd'.
===============================================================================
|    Register Name    |   Type    | Width | Bus | MB | AR | AS | SR | SS | ST |
===============================================================================
|    pc_state_reg     | Flip-flop |  10   |  N  | N  | ?  | ?  | ?  | ?  | ?  |
===============================================================================

Current design is now 'ahtout'
1

/* set_operating_conditions wccom */

/* create_clock clk_xmit  -period 25 */

create_clock Clk_aht_out -period 20
Performing create_clock on port 'Clk_aht_out'.
1
/* otherwise, the delays would exceed the 25 ns! */

compile -map_effort medium

  Loading target library 'lca300kv'
Warning: Multibit library cell 'FD2X4' will not be used in multibit optimization. (OPT-918)
Warning: Multibit library cell 'FD2X2' will not be used in multibit optimization. (OPT-918)
Warning: Multibit library cell 'FD4X4L' will not be used in multibit optimization. (OPT-918)
Warning: Multibit library cell 'LD1X4' will not be used in multibit optimization. (OPT-918)
Warning: Multibit library cell 'FD3X4L' will not be used in multibit optimization. (OPT-918)
Warning: Multibit library cell 'FD3X4' will not be used in multibit optimization. (OPT-918)
Warning: Multibit library cell 'FD3X2' will not be used in multibit optimization. (OPT-918)
```

```
Warning: Multibit library cell ’FD2X4L’ will not be used in multibit optimization. (OPT-918)
Warning: Multibit library cell ’FD1X4’ will not be used in multibit optimization. (OPT-918)
Warning: Multibit library cell ’FD1X2’ will not be used in multibit optimization. (OPT-918)
Warning: Multibit library cell ’FD4X4’ will not be used in multibit optimization. (OPT-918)
Warning: Multibit library cell ’FD4X2’ will not be used in multibit optimization. (OPT-918)
Warning: Multibit library cell ’LD1X4P’ will not be used in multibit optimization. (OPT-918)
Warning: Multibit library cell ’FD1X4L’ will not be used in multibit optimization. (OPT-918)
Warning: Library cell ’FDN1S’ could not be modeled for sequential mapping. (OPT-1201)
Warning: Library cell ’FDN1SP’ could not be modeled for sequential mapping. (OPT-1201)
  Loading design ’ahtout’

  Beginning Resource Allocation  (constraint driven)
  -----------------------------
  Structuring ’ahtout’
  Mapping ’ahtout’
  Allocating blocks in ’ahtout’
  Allocating blocks in ’ahtout’

  Beginning Mapping Optimizations  (Medium effort)
  -------------------------------
  Structuring ’ahtout’
  Mapping ’ahtout’
Information: Changed wire load model for ’ahtout’ from ’B0X0’ to ’B2X2’. (OPT-170)

                                               TOTAL NEG     DESIGN RULE
            TRIALS      AREA     DELTA DELAY     SLACK          COST
           --------    ------    -----------    -------        ------
         0
   --------
         0


  Beginning Phase 1 Design Rule Fixing
  -----------------------------------

                                               TOTAL NEG     DESIGN RULE
            TRIALS      AREA     DELTA DELAY     SLACK          COST
           --------    ------    -----------    -------        ------
         0
   --------
         0


  Beginning Area-Recovery Phase  (cleanup)
  -----------------------------

                                               TOTAL NEG     DESIGN RULE
            TRIALS      AREA     DELTA DELAY     SLACK          COST
           --------    ------    -----------    -------        ------
         4     2741.4          0.00         0.0           0.0
         3     2732.0          0.00         0.0           0.0

.. trials removed. w.l. ...

        19     2001.7          0.00         0.0           0.0
         0
   --------
      3538


  Optimization Complete
  ---------------------
  Transferring Design ’ahtout’ to database ’ahtout.db’
Current design is ’ahtout’.
1

report_area
Information: Updating design information... (UID-85)

****************************************
Report : area
Design : ahtout
Version: 1998.08
Date   : Tue Nov 23 14:58:43 1999
****************************************

Library(s) Used:

    lca300kv (File: /afs/informatik.uni-tuebingen.de/ti/synopsys/1998.8/libraries/syn/lca300kv.db)

Number of ports:               48
Number of nets:               557
Number of cells:              471
Number of references:          22

Combinational area:        536.000000
Noncombinational area:     700.000000
Net Interconnect area:     765.703064

Total cell area:          1236.000000
Total area:               2001.703125
1
```

```
report_timing

****************************************
Report : timing
        -path full
        -delay max
        -max_paths 1
Design : ahtout
Version: 1998.08
Date   : Tue Nov 23 14:58:43 1999
****************************************

Operating Conditions: NOM   Library: lca300kv
Wire Loading Model Mode: enclosed

Design           Wire Loading Model      Library
------------------------------------------------
ahtout                 B2X2              lca300kv


  Startpoint: pc_state_reg<15>
              (rising edge-triggered flip-flop clocked by Clk_aht_out)
  Endpoint: hilf_reg<0>
            (rising edge-triggered flip-flop clocked by Clk_aht_out)
  Path Group: Clk_aht_out
  Path Type: max

  Point                                    Incr       Path
  -----------------------------------------------------------
  clock Clk_aht_out (rise edge)            0.00       0.00
  clock network delay (ideal)              0.00       0.00
  pc_state_reg<15>/CP (FD1)                0.00       0.00 r
  pc_state_reg<15>/QN (FD1)                0.99       0.99 f
  U374/Z (ND2)                             0.47       1.46 r
  U600/Z (OR2P)                            0.44       1.91 r
  U369/Z (ND2)                             0.40       2.31 f
  U578/Z (AO7)                             0.60       2.91 r
  U382/Z (ND2)                             0.46       3.37 f
  U446/Z (IVP)                             0.13       3.51 r
  U414/Z (ND2)                             1.36       4.86 f
  U445/Z (IVP)                             0.34       5.20 r
  U591/Z (IVP)                             0.38       5.58 f
  U444/Z (AO4)                             0.27       5.85 r
  hilf_reg<0>/D (FD1)                      0.00       5.85 r
  data arrival time                                   5.85

  clock Clk_aht_out (rise edge)           20.00      20.00
  clock network delay (ideal)              0.00      20.00
  hilf_reg<0>/CP (FD1)                     0.00      20.00 r
  library setup time                      -0.23      19.77
  data required time                                 19.77
  -----------------------------------------------------------
  data required time                                 19.77
  data arrival time                                  -5.85
  -----------------------------------------------------------
  slack (MET)                                        13.92

1

compile -map_effort medium -boundary_optimization

  Loading design 'ahtout'

  Beginning Mapping Optimizations  (Medium effort)
  -------------------------------
  Structuring 'ahtout'
  Mapping 'ahtout'

                                               TOTAL NEG     DESIGN RULE
            TRIALS      AREA     DELTA DELAY     SLACK          COST
           --------    ------    -----------    -------        ------
         0
   --------
         0


  Beginning Phase 1 Design Rule Fixing
  ------------------------------------

                                               TOTAL NEG     DESIGN RULE
            TRIALS      AREA     DELTA DELAY     SLACK          COST
           --------    ------    -----------    -------        ------
         0
   --------
         0


  Beginning Area-Recovery Phase  (cleanup)
  -----------------------------

                                               TOTAL NEG     DESIGN RULE
            TRIALS      AREA     DELTA DELAY     SLACK          COST
           --------    ------    -----------    -------        ------
         3     2800.8          0.00         0.0           0.0
         2     2795.1          0.00         0.0           0.0

.. trials removed. w.l. ..

        19     2047.7          0.00         0.0           0.0
         0
   --------
      3187


  Optimization Complete
  ---------------------
  Transferring Design ’ahtout’ to database ’ahtout.db’
Current design is ’ahtout’.
1

report_area
Information: Updating design information... (UID-85)

****************************************
Report : area
Design : ahtout
Version: 1998.08
Date   : Tue Nov 23 14:59:27 1999
****************************************

Library(s) Used:

    lca300kv (File: /afs/informatik.uni-tuebingen.de/ti/synopsys/1998.8/libraries/syn/lca300kv.db)

Number of ports:               48
Number of nets:               565
Number of cells:              503
Number of references:          22

Combinational area:        563.000000
Noncombinational area:     700.000000
Net Interconnect area:     784.687439

Total cell area:          1263.000000
Total area:               2047.687500
1

report_timing

****************************************
Report : timing
        -path full
        -delay max
        -max_paths 1
Design : ahtout
Version: 1998.08
Date   : Tue Nov 23 14:59:27 1999
****************************************

Operating Conditions: NOM   Library: lca300kv
Wire Loading Model Mode: enclosed

Design           Wire Loading Model      Library
------------------------------------------------
ahtout                 B2X2              lca300kv

  Startpoint: pc_state_reg<16>
              (rising edge-triggered flip-flop clocked by Clk_aht_out)
  Endpoint: hilf_reg<0>
            (rising edge-triggered flip-flop clocked by Clk_aht_out)
  Path Group: Clk_aht_out
  Path Type: max

  Point                                    Incr       Path
  -----------------------------------------------------------
  clock Clk_aht_out (rise edge)            0.00       0.00
  clock network delay (ideal)              0.00       0.00
  pc_state_reg<16>/CP (FD1)                0.00       0.00 r
  pc_state_reg<16>/QN (FD1)                0.95       0.95 f
  U1012/Z (ND2)                            0.53       1.48 r
  U752/Z (AO4)                             0.55       2.03 f
  U994/Z (AO7)                             0.72       2.75 r
  U732/Z (ND2)                             0.41       3.17 f
  U1009/Z (IVP)                            0.32       3.48 r
  U1007/Z (IVP)                            0.21       3.69 f
  U1008/Z (IVP)                            0.46       4.15 r
  U968/Z (ND2)                             0.34       4.49 f
  U966/Z (AO7)                             0.20       4.70 r
  hilf_reg<0>/D (FD1)                      0.00       4.70 r
  data arrival time                                   4.70

  clock Clk_aht_out (rise edge)           20.00      20.00
  clock network delay (ideal)              0.00      20.00
  hilf_reg<0>/CP (FD1)                     0.00      20.00 r
  library setup time                      -0.20      19.80
  data required time                                 19.80
  -----------------------------------------------------------
  data required time                                  19.80
  data arrival time                                   -4.70
  -----------------------------------------------------------
  slack (MET)                                         15.11

1
```

9 References

References

[Gaetal96] Daniel D. Gajski et al., *System Design Methodologies: Aiming at the 100 h Design Cycle.* IEEE Transaction on VLSI Systems Vol. 4. No. 1. March 1996.

[LaRo97] W. Lange, W. Rosenstiel, *Modellierung einer ATM-Switch-Steuerung*, Internal Report Wilhelm-Schickard-Institut Universität Tübingen WSI 97-6 1997, ISSN 0946-3852 (in German).

[LaRo99-1] W. Lange, W. Rosenstiel, *High-Level Synthese einer ATM-Switch-Steuerung mit dem Behavioral Compiler*$^{\text{TM}}$ *von Synopsys*$^{\text{TM}}$, Internal Report Wilhelm-Schickard-Institut Universität Tübingen WSI 99-1 1999, ISSN 0946-3852 (in German).

[LaRo99-2] W. Lange, W. Rosenstiel, *VHDL Design and High-Level Synthesis of an ATM Switch Controller*, Proceedings of FDL'99 in Lyon, France, pp. 207-216, Sig-VHDL & ECSI 1999. ISBN 2-84010-033-9

[Pryk93] M. D. Pryker, *Asynchronous Transfer Mode: Solution for Broadband ISDN.* Ellis Horwood Ltd, 1993.

[SynPcUG98] *Synopsys Protocol Compiler User Guide.* Version 1998.08.

[SynCDPC98] *Controller Design Using Protocol Compiler* Version 1998.08.

[DaTut97] *Dali Tutorial for VHDL Users.* Version 1997.08.