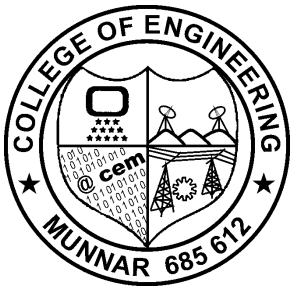
LAB REPORT



04 EC 6593 DESIGN LAB - I

Lab Report submitted in partial fulfillment of the curriculum prescribed for the award of Master of Technology degree in

Electronics & Communication Engineering

with specialization in

VLSI and Embedded Systems

by

4JC11EC0	Student-1
4JC11EC0	Student-2
4JC11EC0	Student-3
4JC11EC0	Student-4

NOVEMBER 2015

COLLEGE OF ENGINEERING MUNNAR

(APJ Abdul Kalam Technological University)

P.B. No. 45, County Hills, Munnar, Kerala - 685612

COLLEGE OF ENGINEERING MUNNAR
P.B. No. 45, County Hills, Munnar - 685612

Affiliated to

APJ Abdul Kalam Technological University

Certificate

This is to certify that the work entitled "04 EC 6593 DESIGN LAB -I" is a bonafide record of work carried out by Student-1 in partial fulfillment of the requirements for the award of the degree of Master of Technology in Electronics & Communication Engineering with specialization in VLSI and Embedded Systems of APJ Abdul Kalam Technological University, during the year 2015-2017.

Staff in Charge
Jayakrishnan K. R.
Asst. Professor
Department of ECE
CoE, Munnar 685612

Head of the Department
Ramesh P.
Associate Professor and Head
Department of ECE
CoE, Munnar 685612

Examiners : 1._____

2._____

Date :

Place : Munnar

3._____

Acknowledgement

"**Acknowledgement** - This is where you thank all those people whom you've rarely (or never) met but whose inspiration, motivation, encouragement, blessings etc. etc. have helped u to write this immensely knowledgeable report and NOT thank your friend who did most of your coding. There is some debate as to whether Abstract should come first or Acknowledgement. I don't care bcz VTU doesnt seem to care. I wrote it this way and it was accepted ... that's all that matters." –**writes Basavraj Talwar** in his website, a jce alumni currently in IISc, who has also written LaTeX-vtu-template

But for preparing this template Anandmurthy Sir of EEE motivated us & gave guidance some suggestions from CRN

And great thanks for Memoir-Document class manual which tells almost all things, which u want while writing a report.

Ashish
Shashi

Table of Contents

List of Figures

List of Tables

Abstract

LaTeX eases our pressure in writing thesis & reports because of its powerful features such as automatic hyphenation, table of contents, figures tables, powerful bibliography tool, citations,Automatic Numbering of Chapter, sections, figures tables, its beautiful fonts, professional output...

Writing too much of code for gives bad impression on LaTeX. But now we have numerous gui tools like gedit-latex-plugin, TeXmaker, Lyx emacs, which are very much user friendly.

This VTU-project-report-template is written using popular document class, “Memoir”. In the coming chapters, we hav given small help manual required for writing report at the end about template

Part I

VLSI Experiments

Chapter 1

Digital Experiments

1.1 Combinational Circuits

1.1.1 Full Adder

Aim

To design and synthesize a full adder using verilog and verify its functionality using RTL and Gate Level Simulation and to draw its layout. Also find the area consumed, power and delay of the full adder.

Program

Full Adder Source

```
/*****************************************************************************
                                    FULL ADDER
*****************************************************************************/
‘timescale 1ns / 1ps
module full_adder(x,y,cin,sum,cout);

        input x,y,cin;
        output sum,cout;

        assign sum=(x^y)^cin;
        assign cout=(x&y)|(x&cin)|(y&cin);

endmodule
```

Full Adder Testbench

```
/*************************************************************************
                                FULL ADDER - TESTBENCH
*************************************************************************/
`timescale 1ns / 1ps
module full_adder_tb;

        // Inputs
        reg x;
        reg y;
        reg cin;

        // Outputs
        wire sum;
        wire cout;

        // Instantiate the Unit Under Test (UUT)
        full_adder uut (
                .x(x),
                .y(y),
                .cin(cin),
                .sum(sum),
                .cout(cout)
        );

        initial begin
                // Initialize Inputs

                //Case 0 : 0+0 with carry in of 0
                x = 0;
                y = 0;
                cin = 0;

                // Wait 100 ns for global reset to finish
                #100;

                // Add stimulus here
```

```
//Case 0 : 0+0 with carry in of 1
x = 0;
y = 0;
cin = 1;
#20;

//Case 0 : 0+1 with carry in of 0
x = 0;
y = 1;
cin = 0;
#20;

//Case 0 : 0+1 with carry in of 1
x = 0;
y = 1;
cin = 1;
#20;

//Case 0 : 1+0 with carry in of 0
x = 1;
y = 0;
cin = 0;
#20;

//Case 0 : 1+0 with carry in of 1
x = 1;
y = 0;
cin = 1;
#20;

//Case 0 : 1+1 with carry in of 0
x = 1;
y = 1;
cin = 0;
#20;

//Case 0 : 1+1 with carry in of 1
x = 1;
y = 1;
```

```
                cin = 1;
                #20;

        end

endmodule
```

Outputs

RTL Simulation

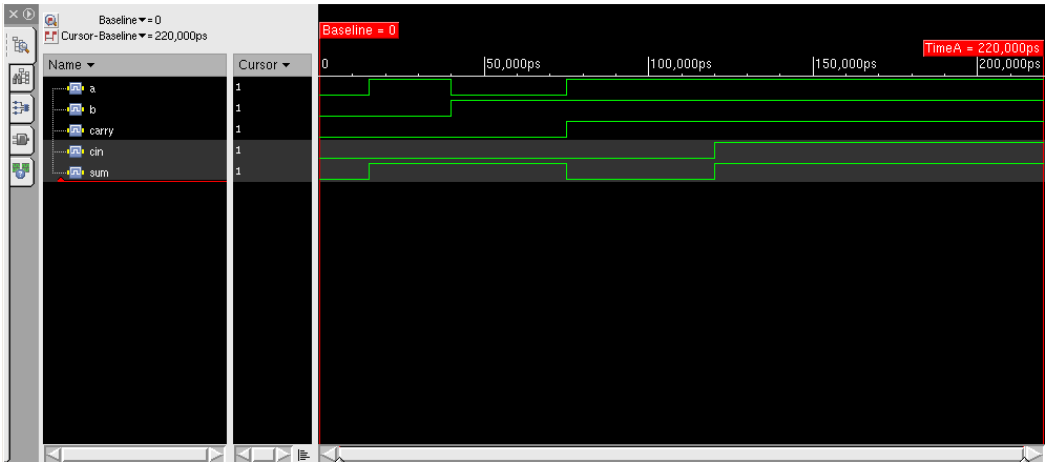


Figure 1.1: RTL Simulation of Full Adder

Synthesized Top Level Schematic

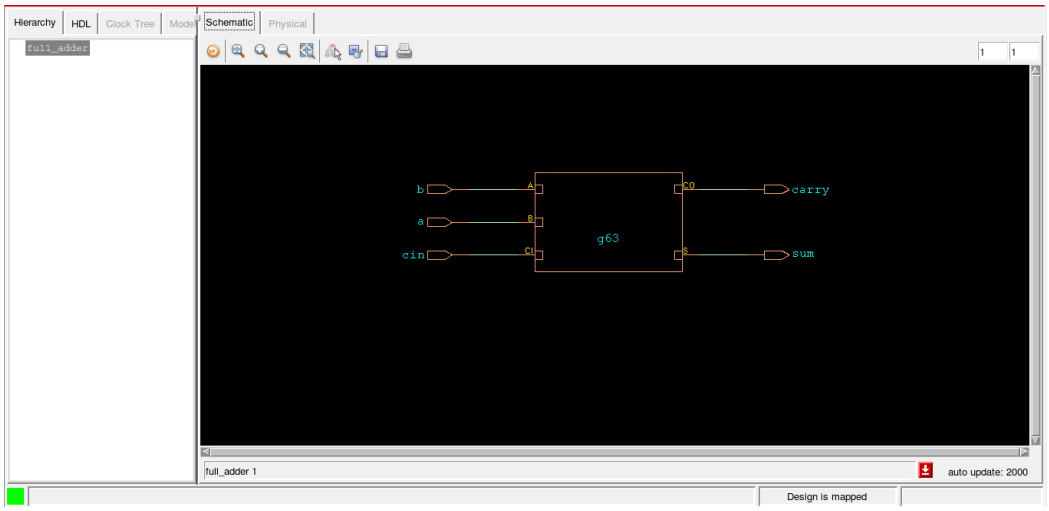


Figure 1.2: Synthesized Top Level Schematic of Full Adder

Total Area consumed

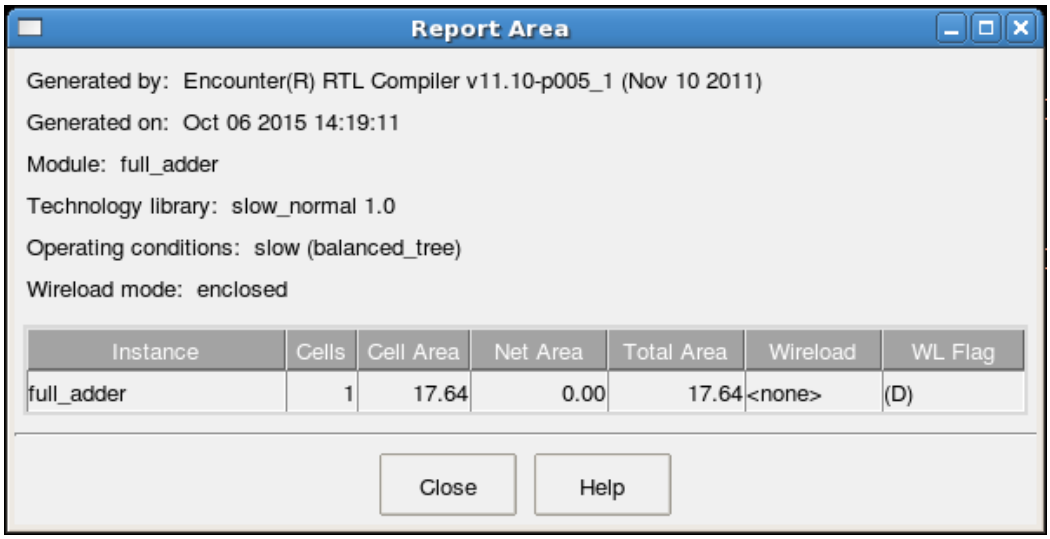


Figure 1.3: Total Area consumed by Full Adder

Total Area consumed

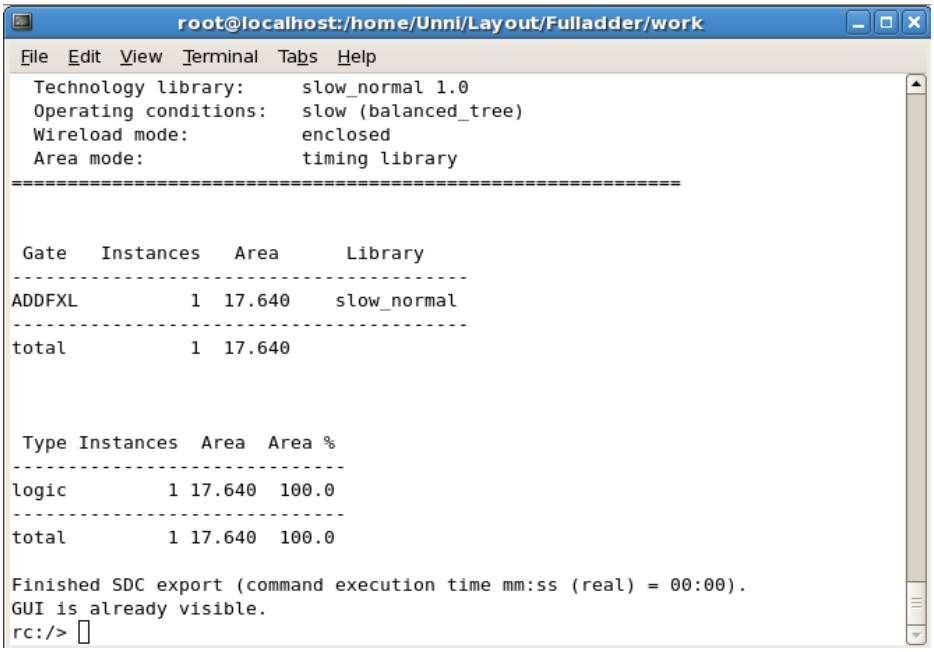


Figure 1.4: Total Area consumed by Full Adder shown in Terminal

Worst Path Delay

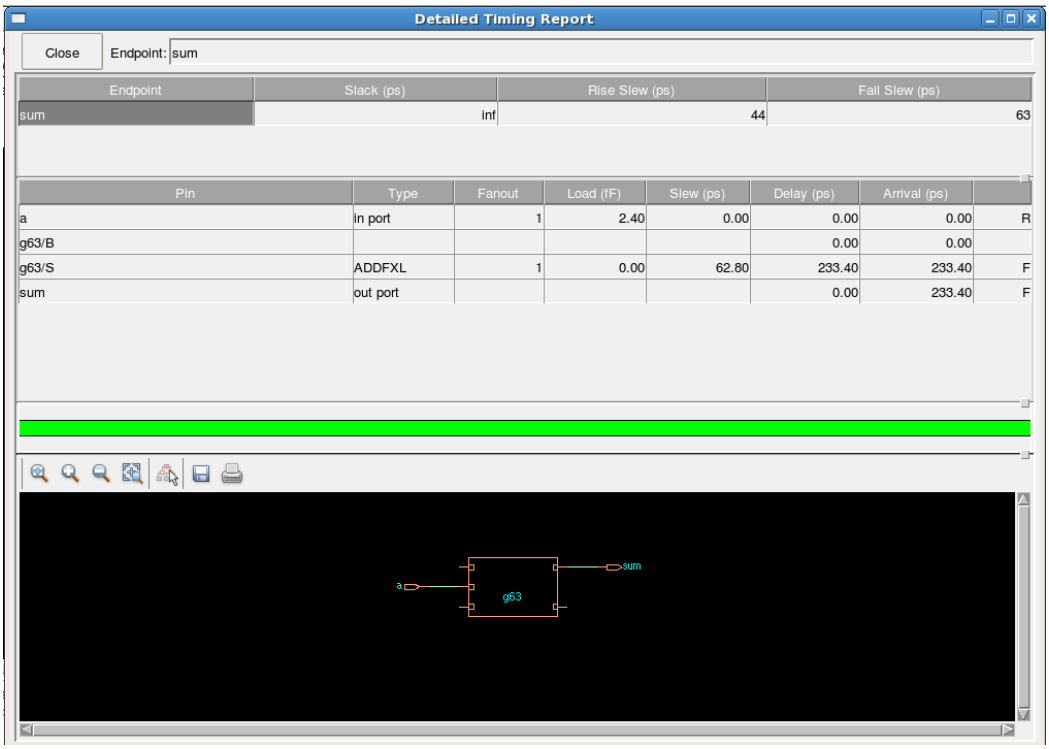


Figure 1.5: Worst Path Delay of Full Adder

Power Consumed

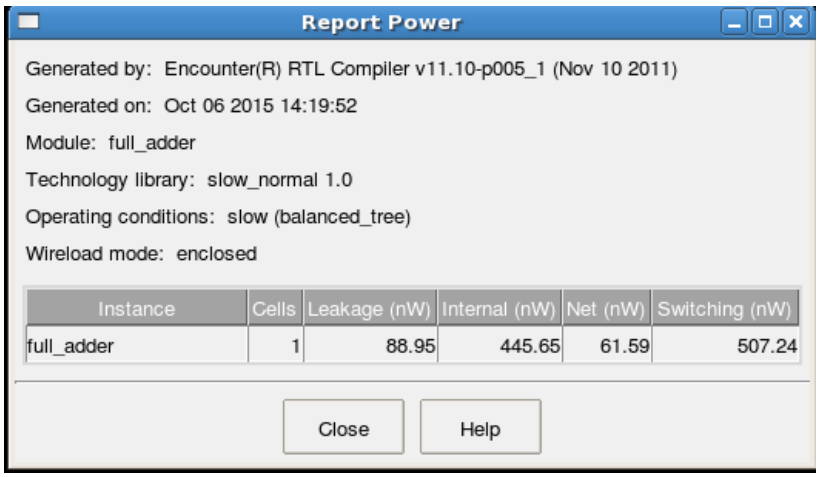


Figure 1.6: Power Consumed by Full Adder

Layout

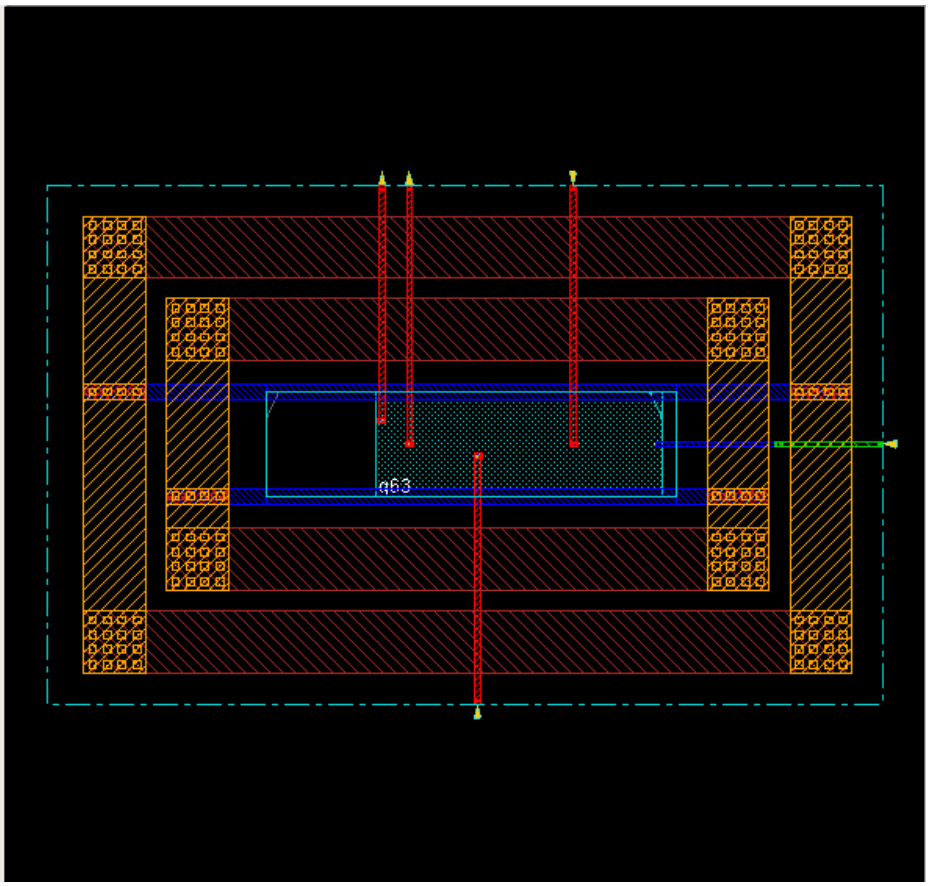


Figure 1.7: Layout of Full Adder

Area Consumed in Layout

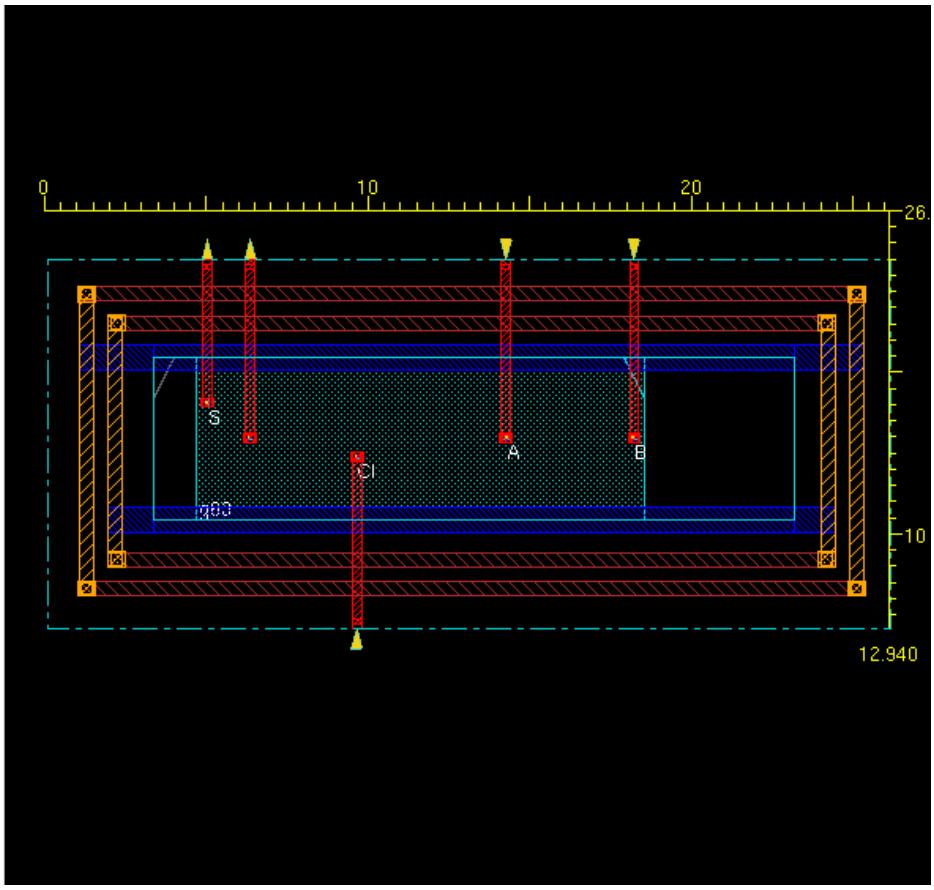


Figure 1.8: Area Consumed by Full Adder in Layout

Gate Level Simulation

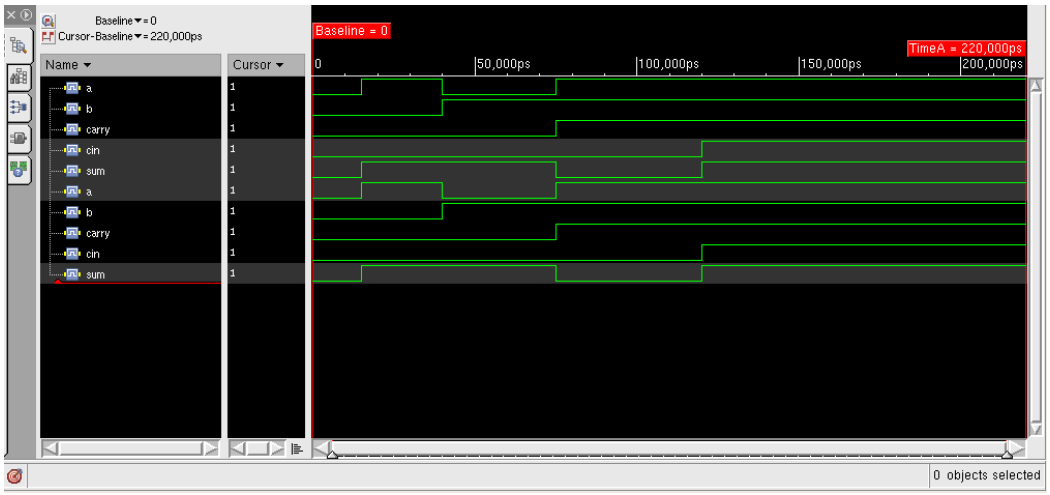


Figure 1.9: Gate Level Simulation of Full Adder

Result

Designed and synthesized a full adder using verilog and verified its functionality using RTL and Gate Level Simulation and also drawn its layout. The area, power and delay of full adder is found out.

1.1.2 Project Organization

The project is split-up into 6 stages as shown in table :

Sl No	Work	Duration(in Weeks)
1	Information Collection on DFB & VNC	1
2	Information collection on development tools	2
3	Integrating DirectFB & VNC	8
4	Cross Compiling & Porting	2
5	Building JPEG libraries	2
6	Testing	2

References

[1] H. Partl:German TEX, *TUGboat* Volume 9, Issue 1 (1988)

[2] Andrew S. Tanenbaum:*Operating Systems Design and Implementation*, Prentice Hall, 2006

[3] About IPTV on Wikipedia `http://en.wikipedia.org/wiki/IPTV`

[4] About VNC on Wikipedia `http://en.wikipedia.org/wiki/Virtual_Network_Computing`

[5] LibVNC server `http://libvncserver.sourceforge.net`

[6] DirectFB documentation `http://elinux.org/DirectFB`

[7] jointSPACE documentation `http://sourceforge.net/apps/mediawiki/jointspace/index.php?title=Main_Page`

[8] PuTTy on Wikipedia `http://en.wikipedia.org/wiki/PuTTy`

[9] Nicola L. C Talbot and Gavin C. Cawley. A fast index assignment algorithm for robust vector quantisation of image data. *In Proceedings of the I.E.E.E. International Conference on Image Processing*, Santa Barbara, California, USA, October 1997.