Mr. Pillem Ramesh, Venkata Aravind Bezawada, K S N Vittal, Dr. Fazal Noorbasha / International Journal of Engineering Research and Applications (IJERA)
ISSN: 2248-9622 www.ijera.com
Vol. 2, Issue 3, May-Jun 2012, pp.1060-1067

Design of 64-bit Peripheral Component Interconnect Bus at 66MHz

Mr. Pillem Ramesh*, Venkata Aravind Bezawada,K S N Vittal***, Dr. Fazal Noorbasha******

ABSTRACT

The original design for the PCI bus was to move high bandwidth peripherals closer to the CPU for performance gains. Graphics-oriented operating systems (OS) and other high bandwidth functions such as Ultra2 SCSI, Fibre Channel, Fast Ethernet, and 3D graphics are consuming more PCI bus bandwidth. This need for more bandwidth has compelled system vendors to find ways of increasing the throughput not only on the PCI bus, but throughout the entire system. The purpose of this paper is the design in 130nm TSMC technology of an 64-bit Peripheral Component Interconnect for its final place and route using SOC Encounter and to make the design to be routable and to meet the timing inorder to work our module at desired frequency by choosing the appropriate Floorplanning for the design by following a standard cell design methodology.

Keywords – **Placement, Routing, PCI bus, SoC Encounter**

I. INTRODUCTION

The SoC Encounter System supports all implementation styles—from flat or hierarchical to single or multi-VDD. The system's fast automatic power grid design and optimization, global routing, in-place optimization, and global timing debug capabilities provide a robust infrastructure to implement any methodology. Full-chip flat prototyping delivers complete and accurate physical, timing, clock, and power data, thereby eliminating the guesswork associated with traditional block-based approaches.

SoC Encounter hierarchical support further helps physical designers to assess how best to partition the logical hierarchy in to physical modules by analyzing the optimal pin assignments; quick time budgeting; accurately predicting the clock distribution networks; analyzing the power grids; and eventually generating complete timing and physical constraints for each of the physical modules.

II. PCI ARCHITECTURE

Since the 32-bit PCI bus is currently able to transfer 133 MB/sec of data less the overhead, what results from changing the bus to 64 bit and 66 MHz? How are PCI bus cycles affected by 64-bit transfers? What are the additions to the PCI bus for 64-bit extensions?

III. PCI BUS CYCLES

Address and data transfers are multiplexed over the same lines on the PCI bus, the address is sent first and then the data. A 32-bit PCI bus has 32 data lines and is able to do 32-bit data transfers and 32-bit memory addressing or 64-bit addressing using two 32-bit PCI cycles known as Dual Address Cycles (DAC).

Memory addressing is not what constrains the PCI bus or system performance. 32-bit addressing allows access to 4 GB of memory--systems such as SMP systems need to address more than this range of memory (see the illustration below). More memory can be addressed with 64-bit addressing in one PCI cycle or two 32-bit cycles using DAC, with the first cycle sending the low address and the second cycle sending the high address.

128 Bytes of Data Being Transferred

	32-bit PCI Bus	64-bit PCI Bus
32-bit Address	1 Cycle 32 Cycles A D	1 Cycle 16 Cycles A D
64-bit Address	2 Cycles 32 Cycles A D	1 Cycle 16 Cycles A D

In the illustration to the right, notice the number of PCI cycles it takes to send the same 128 bytes of data over a 32-bit PCI bus versus a 64-bit bus, assuming the PCI bus is not interrupted. 64-bit PCI bus transactions are more efficient, both for addressing and data, because the number of PCI cycles is reduced to half.

Fig.1: Bus Cycles

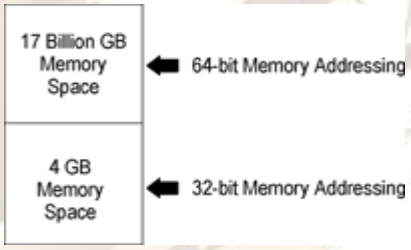


Fig.2: Memory Addressing

IV. 64-BIT EXTENSION PINS

A 64-bit extension to the 32-bit bus architecture requires an additional 39 signal pins which are AD[64::32], C/BE[7::4]#, REQ64#, ACK64#, and PAR64. As we said before, data and addressing are multiplexed over the same pins, either AD[31::00] for 32 bits or AD[31::00] and AD[64::32] for 64 bits. During an address phase, AD[64::32] is used to send the upper 32-bits of a 64-bit address or during a data phase, an additional 32-bits of data. To transfer a 64-bit address in one PCI cycle using the 64-bit bus, you must use the DAC command and assert REQ64#. (Asserting or deasserting a signal means that a particular message is present or missing on the line.) To

transfer the additional 32-bits of data on AD[64::32], REQ64# and ACK64# must be asserted.

The Bus Command and Byte Enable Commands are multiplexed over C/BE[7::4]# pins; Bus Commands are transferred during the address phase and Byte Enable Commands during the data phase. An even parity bit, PAR64, protects the AD[63::32] pins from data corruption. PAR64 has the same timing as AD[63::32] but is delayed by one clock cycle.

64-bit transactions are negotiated using a transaction between a master and a target asserting REQ64# and ACK64#. Devices determine if they are connected to a 64-bit bus by asserting or deasserting REQ64# when RST# is deasserted. Only memory commands support 64-bit data transfers.

V. 64-BIT BUS BENEFITS

Many factors play into overall system performance and affect the industry's progress to 64-bit PCI. The 32-bit PCI bus is not in itself slowing system performance. Peripheral devices such as SCSI, IDE, and Fast Ethernet by themselves do not use the full potential of the current PCI bus. Interaction between the PCI bus, the Host Bridge, DRAM, and the CPU commonly slow down PCI transfers.

Let's also consider the transactions in the system. The CPU communicates with Dynamic Random Access Memory (DRAM) and the Host Bridge: the Host Bridge in turn communicates with the PCI bus and DRAM. Direct Memory Access (DMA) devices transfer data directly to DRAM through the PCI bus as shown here.

The burst rate of data throughput on the PCI bus doubles with 64-bit data transfers. 64-bit DMA devices can move data in 64-bit chunks directly between the PCI bus and DRAM if the PCI bus and DRAM are set up the handle 64-bit transfers. The system with a 64-bit PCI bus is less congested; 64-bit devices in the system get on and off the bus in half the time, making the PCI bus more efficient. In essence, since 64-bit transfers achieve better PCI bus utilization and more devices can be added to the bus before realizing the bus' full bandwidth potential. The more heavily weighted your system becomes with peripheral devices the more it benefits from a 64-bit wide PCI bus.

Notice in the following drawings, the number of components that would be affected by increasing the PCI bus to 64-bits, the host bridge, the DRAM, the CPU and even the OS and driver. Since the majority of transactions are data transfers from the PCI bus to DRAM and from DRAM to the PCI bus, you could consider that increasing the bandwidth of the bus would increase the performance of the system.

However, the CPU and the OS may become the bottleneck even if implementing a 64-bit PCI bus.

VI. PCI BUS SPEEDS 33MHZ VS 66MHZ

Another way of increasing the throughput of the PCI bus is increasing the PCI clock speed. PCI systems are now at 33 MHz; the PCI spec defines 66 MHz as a way of increasing PCI bandwidth. 66-MHz devices are great for high bandwidth applications and peripherals. Just as 64-bit architecture can double the bus bandwidth, 66 MHz can double the throughput. The diagram to the right illustrates the increase in throughput.

Theoretical Throughput

	32-bit PCI Bus	64-bit PCI Bus
33 Mhz	133 MB/sec	266 MB/sec
66 Mhz	266 MB/sec	532 MB/sec

Some hardware modifications need to be made to the PCI devices and motherboard to allow this increase in clock speed. 66-MHz devices are defined by modifying an existing ground pin to a static signal (M66EN) using a single pullup resistor and adding one bit to the Configuration Status register. 66-MHz PCI requires higher maximum clock frequency and modifying timing parameters. Engineers need to pay close attention to maximum trace lengths, loading of add-in boards, and the maximum pin capacitance of all add-in boards. A 66-MHz bus is capable of operating at 0 to 66 MHz speeds. 33-MHz devices operate at 33 MHz in a 66-MHz bus; likewise 66-MHz devices operate at 33 MHz in a 33-MHz bus.

When designing a 66-MHz PCI bus into the motherboard, electrical problems, chip size, and heat dissipation are some issues to deal with. The loading factor in the 66-MHz bus is significantly reduced from that of the 33 MHz. As the bus speed increases, the total distance data can travel decreases, and the number of slots available on the motherboard decreases. To solve the 66-MHz requirements on a motherboard, 66-MHz PCI bus is typically separated out on another bus using a 66-MHz PCI bridge chip.

VII. 64-BIT MARKET TRENDS

System vendors are looking for ways of creating the fastest and best systems on the market. Who are some of the players in this market segment? How do the OS, the CPU, and other system components affect system performance?

1. PLACEMENT AND ROUTING

1.1. DESIGN IMPORT

The files that will be imported in to the SoC Encounter Tool to start place and route are the netlist file, timing library files, library exchange format files, timing constraints file and I/O assignment file.

After Importing the above files in to SoC Encounter we perform sanity checks. We check for whether any floating input pins, any multidriven ports and nets, any nets with tristate drivers, if any input delay or load information in sdc is missing, if there are any undefined clocks, whether any don't use cells in the design, whether there are any negative slack paths after synthesis. So after performing these checks the timing report before placing the standard cells is shown below.

```
------------------------------------------------------------
           timeDesign Summary
------------------------------------------------------------

+--------------------+---------+---------+---------+---------+
-+
|     Setup mode     |   all   | reg2reg | in2reg  | reg2out |
|
+--------------------+---------+---------+---------+---------+
-+
|           WNS (ns):| -0.474  |  2.160  | -0.474  |  0.004  |
|
|           TNS (ns):| -0.573  |  0.000  | -0.573  |  0.000  |
|
|    Violating Paths:|    3    |    0    |    3    |    0    |
|
|          All Paths:|   5447  |   5397  |   104   |   47    |
|
+--------------------+---------+---------+---------+---------+
-+

Density: 0%
------------------------------------------------------------
```

From the report it is clear that there are three violating paths under input to register timing paths. We can still go ahead and do the floor planning because the tool can easily fix them by upsizing the cells .

1.2. FLOOR PLANNING

In floor planning step we decide how much area will be required for our design to place in the core area so that minimum wire length will be used to make connectivity between standard cells. We also create the input output pins.

The process of arranging the macros in a block or chip is called floor planning . This step is critical as a bad floor panning can result in a design which can be congested or which cannot be closed in terms of timing

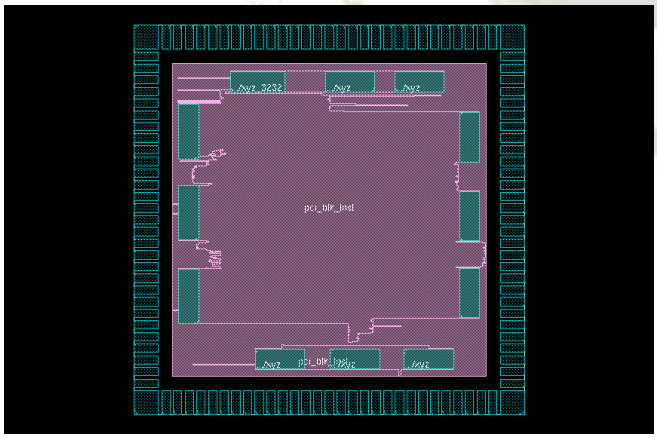


Fig.3: Amoeba view for Floor Plan of PCI bus

1.3. POWER PLANNING

The core-utilization is 32.5%. This step involves the creation of power stripes and core ring at the chip level. The aim of power planning is to achieve minimum IR drop.

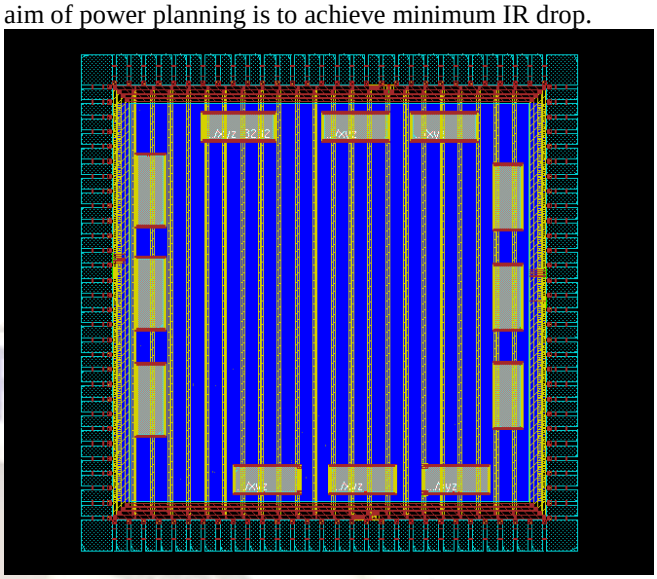
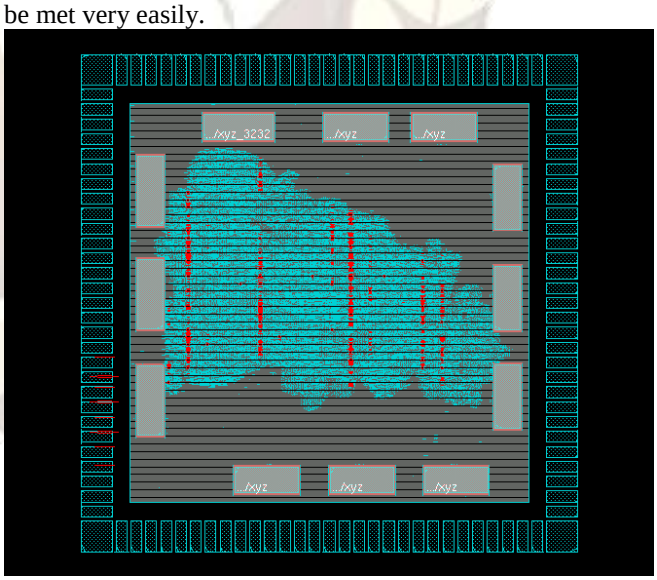


Fig.4. Power planning of PCI bus.

1.4. PLACEMENT

In this step we place the standard cells . Before the placement of standard cells we place the endcap cells to the left and right of the core so as to avoid process antenna violations and spacing violations. The simple idea is to minimize the length of wires, but, for example, in a timing-driven placement the intention is to decrease as much as possible the delay on the critical paths. So that timing will be met very easily.



From the figure we can observe diamond shaped red color boxes indicating the horizontal or vertical congestion. This is mainly because of the placement of standard cells below the metal-4 vertical power stripe. Now if we check for timing the tool will do a trail route automatically and gives the report. The term WNS indicates Worst Negative Slack on a particular path and TNS indicates Total Negative Slack.

The timing report shown for this design at this stage is

timeDesign Summary

Setup mode	all	reg2reg	in2reg	reg2out
WNS (ns):	-0.614	1.674	-0.614	-0.417
TNS (ns):	-2.675	0.000	-2.259	-0.417
Violating Paths:	8	0	7	1
All Paths:	5447	5397	104	47

DRVs	Real		Total
	Nr nets(terms)	Worst Vio	Nr nets(terms)
max_cap	0 (0)	0.000	35 (35)
max_tran	19 (19)	-1.350	19 (19)
max_fanout	0 (0)	0	4 (4)

Density: 23.799%
Routing Overflow: 0.00% H and 0.34% V

So in order to remove the congestion we have to declare placement blockage on metal-4. So the standard cells will not be placed below the metal-4 stripe as shown below.

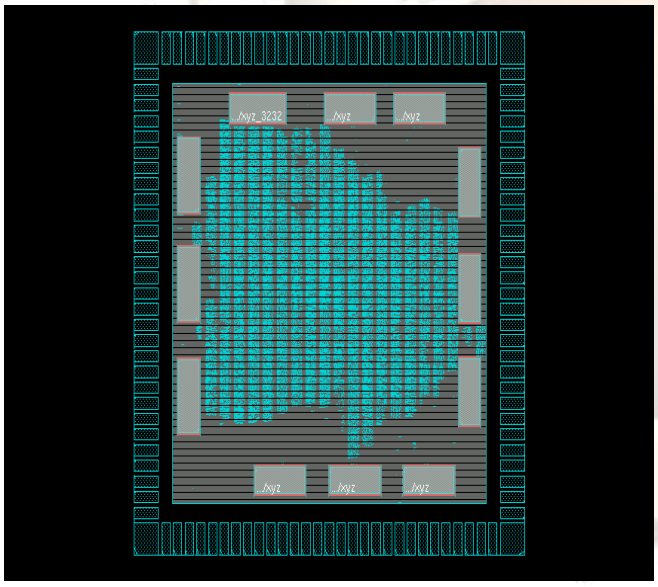


Fig.5. Placement with out congestion

Now if we check for timing the tool will do a trail route automatically and gives the report.

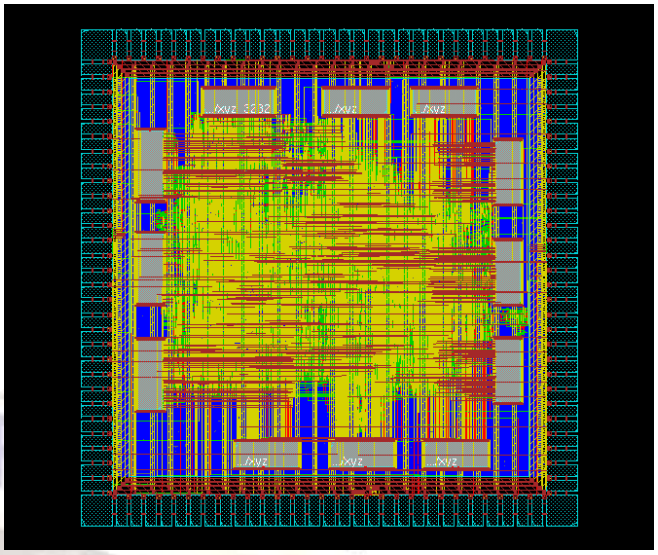


Fig.6. Trail Route

timeDesign Summary

Setup mode	all	reg2reg	in2reg	reg2out
WNS (ns):	-6.968	-6.968	-0.662	-0.364
TNS (ns):	-475.044	-467.879	-6.316	-0.850
Violating Paths:	361	317	40	4
All Paths:	5447	5397	104	47

DRVs	Real		Total
	Nr nets(terms)	Worst Vio	Nr nets(terms)
max_cap	3 (3)	-15.315	38 (38)
max_tran	38 (744)	-38.361	38 (744)
max_fanout	117 (117)	-2141	121 (121)

Density: 28.015%
Routing Overflow: 0.00% H and 0.00% V

We can clearly see that the congestion is 0% which is good and we can go ahead to the next step for placement and routing.

1.5 POST-PLACEMENT OPTIMIZATION

After optimizing the design for removal of negative slack the density of the design has changed to 24.837%. The timing report is shown below :

```
------------------------------------------------------------
     optDesign Final Summary
------------------------------------------------------------

+--------------------+---------+---------+---------+---------+
|     Setup mode     |   all   | reg2reg | in2reg  | reg2out |
+--------------------+---------+---------+---------+---------+
|           WNS (ns):|  0.105  |  0.280  |  0.105  |  0.142  |
|           TNS (ns):|  0.000  |  0.000  |  0.000  |  0.000  |
|    Violating Paths:|    0    |    0    |    0    |    0    |
|          All Paths:|  5447   |  5397   |   104   |   47    |
+--------------------+---------+---------+---------+---------+

+---------------+-------------------------------+-----------------+
|               |             Real              |       Total     |
|    DRVs       +-----------------+-------------+-----------------|
|               |  Nr nets(terms) | Worst Vio   |  Nr nets(terms) |
+---------------+-----------------+-------------+-----------------+
|   max_cap     |      0 (0)      |   0.000     |     35 (35)     |
|   max_tran    |      0 (0)      |   0.000     |      0 (0)      |
|   max_fanout  |    325 (325)    |    -59      |    329 (329)    |
+---------------+-----------------+-------------+-----------------+

Density: 24.837%
Routing Overflow: 0.00% H and 0.00% V
```

Now the timing violations and drv violations have been eliminated after optimization and setup slack has become positive.

Post-Placement optimization can also be called as Pre-CTS optimization.

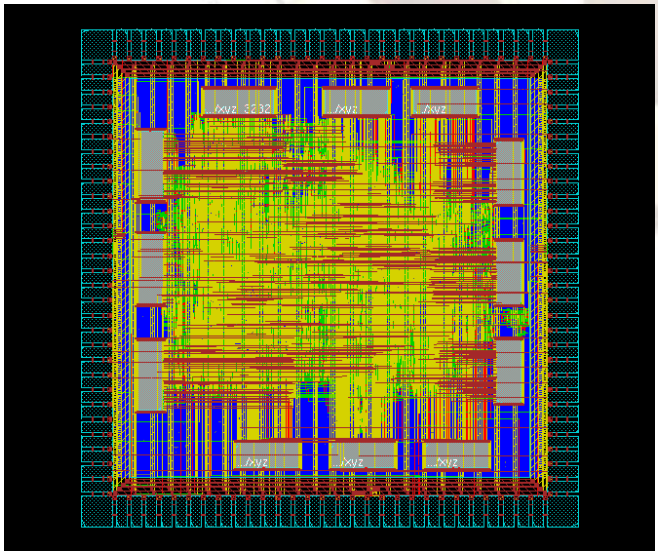


Fig.7. Pre-CTS optimization

1.6. CLOCK TREE SYNTHESIS

The next step is generation of clock tree. It is termed as Clock Tree Synthesis (CTS). How the clock is distributed in a design is one of the most limiting factors in order to achieve high frequency circuits. The clock signal has to get all the clocked components at the same time in order to achieve a correct operation, and the more the frequency is increased, the more clock inaccuracy we have. The generated clock tree is highlighted in white lines as shown in Figure 8. The density of the design after clock tree synthesis is 25.813%.

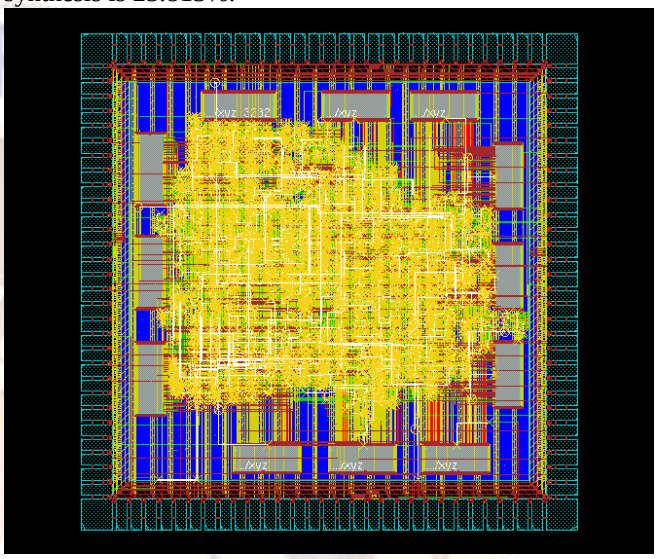


Fig.8: Clock Tree Synthesis

The timing report at this stage is shown below:

```
+--------------------+---------+---------+---------+---------+
|     Setup mode     |   all   | reg2reg | in2reg  | reg2out |
+--------------------+---------+---------+---------+---------+
|           WNS (ns):| -0.475  |  0.281  |  0.715  | -0.475  |
|           TNS (ns):| -0.475  |  0.000  |  0.000  | -0.475  |
|    Violating Paths:|    1    |    0    |    0    |    1    |
|          All Paths:|  5447   |  5397   |   104   |   47    |
+--------------------+---------+---------+---------+---------+
```

DRVs	Real		Total
	Nr nets(terms)	Worst Vio	Nr nets(terms)
max_cap	0 (0)	0.000	33 (33)
max_tran	0 (0)	0.000	0 (0)
max_fanout	325 (325)	-59	448 (448)

Density: 25.813%
Routing Overflow: 0.00% H and 0.02% V

The negative setup slack seen in the report is mainly because during clock tree generation tool adds buffers to all the clock nets. This increases the path delay. So to eliminate this negative slack we optimize the design.

1.7. POST-CTS OPTIMIZATION

This step is referred to as the Post-CTS optimization. After CTS we can also check for the hold violations. But in most of the cases we perform hold check after detailed routing only. Even if hold check is done at CTS stage we can easily fix the hold violations during Post-CTS optimization.

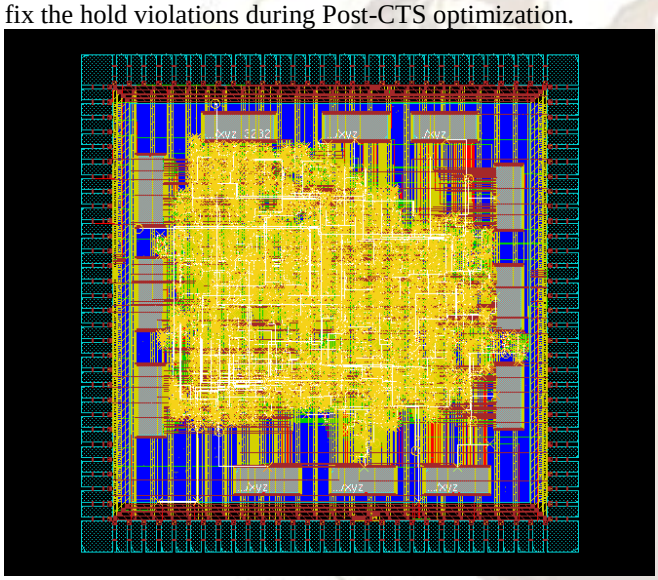


Fig.9: Post-CTS optimization

The Density of the design has become 25.799%. The timing report is shown below:

optDesign Final Summary

Setup mode	all	reg2reg	in2reg	reg2out
WNS (ns):	0.049	0.286	0.111	0.049
TNS (ns):	0.000	0.000	0.000	0.000
Violating Paths:	0	0	0	0
All Paths:	5447	5397	104	47

DRVs	Real		Total
	Nr nets(terms)	Worst Vio	Nr nets(terms)
max_cap	0 (0)	0.000	33 (33)
max_tran	0 (0)	0.000	0 (0)
max_fanout	325 (325)	-59	448 (448)

Density: 25.799%
Routing Overflow: 0.00% H and 0.02% V

After Post-CTS optimization it is clear that the setup violation is eliminated and there are no violations on any other paths .

1.8. DETAILED ROUTING

The next step is detailed routing. Here the tool does the actual timing driven sign-off routing. The trail route between the standard cells is removed.

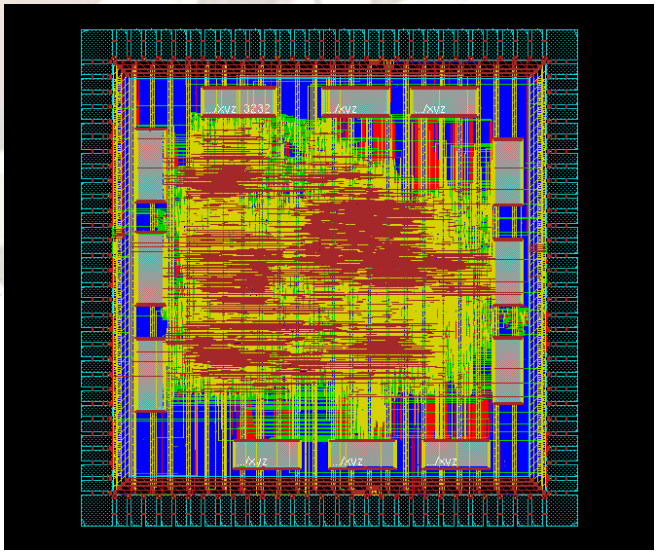


Fig.10: Detailed Routing

The density of the design is at 25.799%. The timing is checked for both setup and hold requirements. There is no need of fixing hold time at early stages of the design.

The timing report is shown below:

```
------------------------------------------------------------
          timeDesign Summary
------------------------------------------------------------

+--------------------+---------+---------+---------+---------+
|     Setup mode     |   all   | reg2reg | in2reg  | reg2out |
+--------------------+---------+---------+---------+---------+
|           WNS (ns):| -0.213  | -0.213  | -0.090  |  0.022  |
|           TNS (ns):| -1.111  | -0.525  | -0.586  |  0.000  |
|    Violating Paths:|   13    |    4    |    9    |    0    |
|          All Paths:|  5447   |  5397   |   104   |   47    |
+--------------------+---------+---------+---------+---------+

+---------------+-----------------------------+-----------------+
|               |            Real             |      Total      |
|    DRVs       +-----------------+-----------+-----------------+
|               | Nr nets(terms)  | Worst Vio | Nr nets(terms)  |
+---------------+-----------------+-----------+-----------------+
|   max_cap     |      0 (0)      |   0.000   |     33 (33)     |
|   max_tran    |      1 (1)      |  -2.884   |      1 (1)      |
|   max_fanout  |    325 (325)    |    -59    |    448 (448)    |
+---------------+-----------------+-----------+-----------------+

Density: 25.799%
```

```
------------------------------------------------------------
          timeDesign Summary
------------------------------------------------------------

+--------------------+---------+---------+---------+---------+
|     Hold mode      |   all   | reg2reg | in2reg  | reg2out |
+--------------------+---------+---------+---------+---------+
|           WNS (ns):|  0.104  |  0.104  |  1.873  |  1.615  |
|           TNS (ns):|  0.000  |  0.000  |  0.000  |  0.000  |
|    Violating Paths:|    0    |    0    |    0    |    0    |
|          All Paths:|  5445   |  5397   |   103   |   46    |
+--------------------+---------+---------+---------+---------+

Density: 25.799%
```

1.9. POST-ROUTE OPTIMIZATION

The hold violations are always fixed at post route optimization because the skews seen are real and to fix them tool can easily add buffers. Before detailed routing we only give priority for fixing the setup timing violations.

During Post-route optimization the tool optimizes the design by upsizing or downsizing the cells or by adding some buffers.

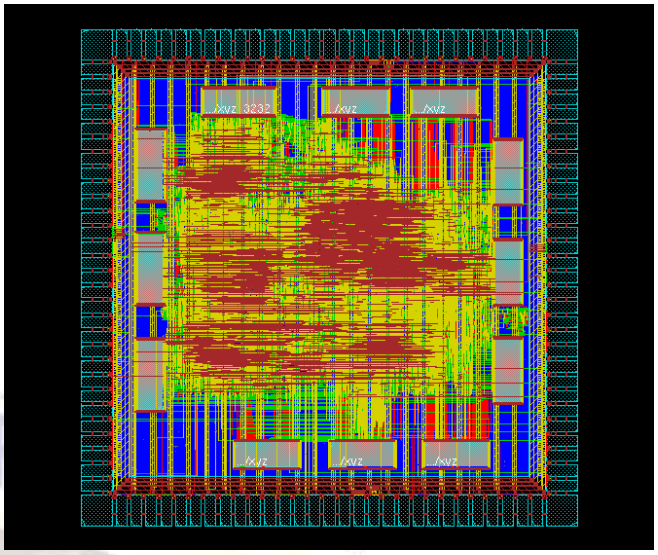


Fig.7: Post-route Optimization

The density of the design is 25.825%.The timing report is shown below:

```
------------------------------------------------------------
     optDesign Final Summary
------------------------------------------------------------

+--------------------+---------+---------+---------+---------+
|     Setup mode     |   all   | reg2reg | in2reg  | reg2out |
+--------------------+---------+---------+---------+---------+
|           WNS (ns):|  0.022  |  0.796  |  0.032  |  0.022  |
|           TNS (ns):|  0.000  |  0.000  |  0.000  |  0.000  |
|    Violating Paths:|    0    |    0    |    0    |    0    |
|          All Paths:|  5447   |  5397   |   104   |   47    |
+--------------------+---------+---------+---------+---------+

+--------------------+---------+---------+---------+---------+
|     Hold mode      |   all   | reg2reg | in2reg  | reg2out |
+--------------------+---------+---------+---------+---------+
|           WNS (ns):|  0.104  |  0.104  |  1.874  |  1.615  |
|           TNS (ns):|  0.000  |  0.000  |  0.000  |  0.000  |
|    Violating Paths:|    0    |    0    |    0    |    0    |
|          All Paths:|  5445   |  5397   |   103   |   46    |
+--------------------+---------+---------+---------+---------+

+---------------+-----------------------------+-----------------+
|               |            Real             |      Total      |
|    DRVs       +-----------------+-----------+-----------------+
|               | Nr nets(terms)  | Worst Vio | Nr nets(terms)  |
+---------------+-----------------+-----------+-----------------+
|   max_cap     |      0 (0)      |   0.000   |     33 (33)     |
|   max_tran    |      0 (0)      |   0.000   |      0 (0)      |
|   max_fanout  |    325 (325)    |    -59    |    448 (448)    |
+---------------+-----------------+-----------+-----------------+

Density: 25.825%
------------------------------------------------------------
```

It is clearly seen that the hold violations on all the paths are removed after Post-route optimization. After this step we can fill the spaces in our design with the filler cells for the continuity of N-well regions to avoid any DRC violations.

VIII. CONCLUSION

Placement and routing of 64-bit PCI bus is done at 130nm technology using SoC Encounter tool version 9.1. The timing reports at each and every step of the design are observed and if any violations they are removed and from the reports finally the setup slack is 0.022ns and the hold slack is 0.104ns. The density of the design is finally reported as 25.825%. The increase in density of the design at each and every step is mainly because of the increment in net lengths ,upsizing of the cells, addition of buffers.The voltage of operation is 1.8v. The power consumption of the chip level PCI bus is 120mW and the total area is 1422162.9μm^2.

REFERENCES

[1] Ababei, C.; Feng, Y.; Goplen, B.; Hushrav Mogal; Zhang, T.; Bazargan, K.; Sachin Sapatnekar; "Placement and routing in 3D integrated circuits", IEEE journal, vol.22, pp.520-531, November 2005.

[2] K. Banerjee, S. J. Souri, P. Kapur, and K. C. Saraswat, "3-D ICs: A novel chip design for improving deep submicron interconnect performance and systems-on-chip integration," Proceedings of the IEEE, vol. 89, pp. 602–633, May 2001.

[3] J. Davis, R. Venkatesan, A. Kaloyeros, M. Beylansky, S. Souri, K. Banerjee, K. Saraswat, A. Rahman, R. Reif, and J. Meindl, "Interconnect limits on gigascale integration (GSI) in the 21st century," Proceedings of the IEEE, vol. 89, pp. 305–324, March 2001.

[4] K. W. Guarini, A. W. Topol, M. Leong, R. Yu, L. Shi, M. R. Newport, D. J. Frank, D. V. Singh, G. M. Cohen, S. V. Nitta, D. C. Boyd, P. A. O'Neil, S. L. Tempest, H. B. Pogpe, S. Purushotharnan, and W. E. Haensch, "Electrical integrity of state-of-the-art 0.13_m SOI CMOS devices and circuits transferred for three-dimensional (3D) integrated circuit (IC) fabrication," in Technical Digest of the IEEE International Electron Devices Meeting, pp. 943–945, 2002.

[5] J. Burns, L. McIlrath, J. Hopwood, C. Keast, D. P. Vu, K.Warner, and P. Wyatt, "An SOI-based three dimensional integrated circuit technology," in IEEE International SOI Conference, pp. 20–21, Oct. 2000.

[6] R. Reif, A. Fan, K. N. Chen, and S. Das, "Fabrication technologies for three-dimensional integrated circuits," in Proceedings of the International Symposium on Quality Electronic Design (ISQED), 2002.

[7] A. J. Alexander, J. P. Cohoon, J. L. Colflesh, J. Karro, E. L. Peters, and G. Robins, "Placement and routing for three-dimensional FPGAs," in Fourth Canadian Workshop on Field-Programmable Devices, pp. 11–18, 1996.

Pillem. Ramesh was born in 1982 at krishna district of Andhra Pradesh state, india. He completed Post Graduation in VLSI System Design from SITAMS, Chittoor, JNTU, Hyderabad. Presently he is Working as Asst.Professor in K. L. University. His interested areas are Analog VLSI circuits and NANO CMOS.

Venkata Aravind Bezawada was born in A.P,India. He received the **B.Tech** degree in **Electronics& communications Engineering** from **Jawaharlal Nehru Technological University** in 2009. Presently he is pursuing **M.Tech VLSI Design** in **K L University**. His research interests include **VLSI Physical Design, Low Power Design.**

K S N Vittal was born in A.P. India. He received the **B.Tech** degree in **Electronics& communications Engineering** from **Jawaharlal Nehru Technological University** in 2008. Presently he is pursuing **M.Tech VLSI Design** in **K L University**. His research interests include **Analog design** and **Low power design.**

Dr. Fazal Noorbasha was born in India. He received **Ph.D.** in **Department of Physics and Electronics** from Dr. HariSingh Gour Central University. Presently he is working as **Assistant Professor**, **Department of E.C.E, K L University.**