

**THE WESTERN DESIGN CENTER, INC.**

2166 E. Brown Rd. Mesa, AZ 85213

Ph 480-962-4545 Fx 480-835-6442

[www.westerndesigncenter.com](http://www.westerndesigncenter.com)

# **W65C02S DESIGN REUSE**

## **INFORMATION and DATA SHEET**

An Introduction to Reuse Methodology with System-On-a-Chip (SOC) Design

**March 2001**

**Copyright © 2001**  
**by**  
**The Western Design Center, Inc.**  
**All Rights Reserved**

**Worldwide**

# TABLE OF CONTENTS

Chapter 1	Introduction to Reuse Methodology with System-On-a-Chip (SOC) Design	5
	1.1 System-On-a-Chip Design	5
Chapter 2	Specification of System-On-a-Chip	9
	2.1 Specification for W65C122S	9
	2.2 Pin Descriptions	11
	2.2.1 Address Bus (A0-A15)	11
	2.2.2 Bus Enable (BE)	11
	2.2.3 Control Lines (CA1, CA2)	11
	2.2.4 Control Lines (CB1, DB2)	11
	2.2.5 Data Bus (D0-D7)	11
	2.2.6 External Memory (EXTMEMB)	11
	2.2.7 FCLK	12
	2.2.8 Interrupt Request (IRQB)	12
	2.2.9 No Connect (NC)	12
	2.2.10 Non-maskable Interrupt (NMIB)	12
	2.2.11 Peripheral Data Port A (PA0-PA7)	12
	2.2.12 Peripheral Data Port B (PB0-PB7)	12
	2.2.13 PH12	12
	2.2.14 Power Supply (VDD, VSS)	13
	2.2.15 Read/Write (RWB)	13
	2.2.16 Ready	13
	2.2.17 Reset	13
	2.2.18 Synchronize (SYNC)	13
Chapter 3	Design Methodology	21
	3.1 Design Tools	21
	3.2 Design Flow for Retargeting	22
	3.2.1 Retargeting flow for MOSIS manufacturing	23

## TABLE OF FIGURES

Chapter 1	Figure 1.1 Flowchart for Design, Manufacturing and Test	6
	Figure 1.2 W65C02 Developer Board	7
	Figure 1.3 W65C134S Footprint	8
	Figure 1.4 W65C122S Footprint	8
Chapter 2	Figure 2.1 W65C02S Status Register Coding	9
	Figure 2.2 W65C02S Programming Model	10
	Figure 2.3 Block Diagram of W65C122S	10
	Figure 2.4 W65C02 GDSII Mask Abstract Cell	16
	Figure 2.5 W65C22 GDSII Mask Abstract Cell	16
	Figure 2.6 W4KROM GDSII Mask Abstract Cell	17
	Figure 2.7 W256RAM GDSII Mask Abstract Cell	17
	Figure 2.8 W65C122S Viewlogic Schematic	18
	Figure 2.9 W65C122SPL Wire Bonding Diagram	19
	Figure 2.10 W65C134SPL Wire Bonding Diagram	20
Chapter 3	Figure 3.1 Design Tools	21
	Figure 3.2 Design Flow for Retargeting	22

## **TABLE OF TABLES**

Chapter 2	Table 2.1 Features of the W65C122S	9
	Table 2.2 System Memory Map	11
	Table 2.3 WDC Standard Cell Library for Mask Design	14
	Table 2.4 WDC Standard Cell Library for Schematic Design	15

## **GLOSSARY**

# Chapter 1

## Introduction to Reuse Methodology with System-On-a-Chip (SOC) Design

Time-to-market and production costs are two critical issues associated with Integrated Circuit (IC) design. When portions of the IC can be reused the time-to-market is reduced. For example, a microprocessor core can be reused in many different IC's. An IC with a MicroProcessor Unit (MPU) Input/Output (I/O) circuitry, data memory (RAM) and program memory (ROM) is called a MicroController Unit (MCU). When a MCU is the only chip in a system it is thought of as a System-On-a-Chip (SOC).

WDC is a pioneer in both microprocessor technology and also in providing microprocessor Intellectual Property (IP) to the industry. WDC is referred to as both a Fabless Semiconductor Company and an IP Provider. WDC is the original source and IP owner of the patented CMOS W65C02<sup>TM</sup> 8-bit MPU and W65C816<sup>TM</sup> 16-bit MPU and has been licensing these MPU cores since 1981.

WDC's microprocessor IP is provided as soft, firm and hard cores. "Soft" cores are the synthesizable Register Transfer Level (RTL) behavioral models in Verilog Hardware Description Language (HDL). A "firm" core can be thought of as a soft core synthesized into a field programmable technology such as FPGA's. A "hard" core is synthesized with an ASIC library of GDSII cells and then placed and routed. Hard cores can be automatically placed and routed or manually placed and routed. Manually synthesized placed and routed hard cores are typically two or more times smaller than automatically placed and routed cores. WDC has manually synthesized, optimized, placed and routed hard cores.

IP is defined as Patents, Copyrights, Maskworks, Trademarks, and Trade Secrets.

### 1.1 System-On-a-Chip Design

The flow chart in Figure 1.1 gives the steps for a MCU SOC design, manufacturing and test.

The W65C122S design can replace WDC's W65C02DB microprocessor development board as a SOC. The pins of the new chip are also compatible with the W65C134S MCU. The block diagram of the W65C02DB Developer Board is given in Figure 1.2. The pin-out or "footprint" for the W65C134S is given in Figure 1.3.

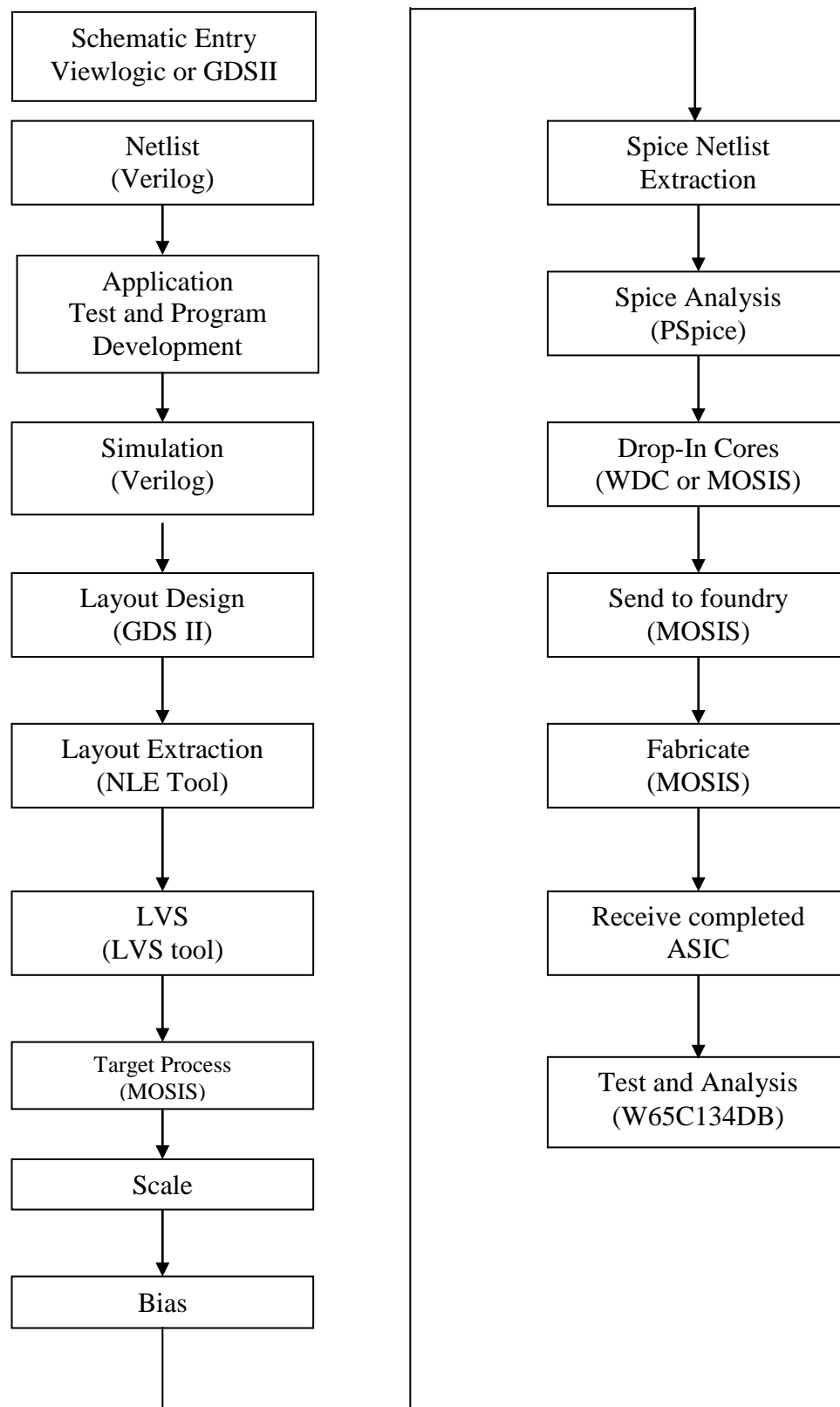


Figure 1.1 Flow Chart for Design, Manufacturing and Test

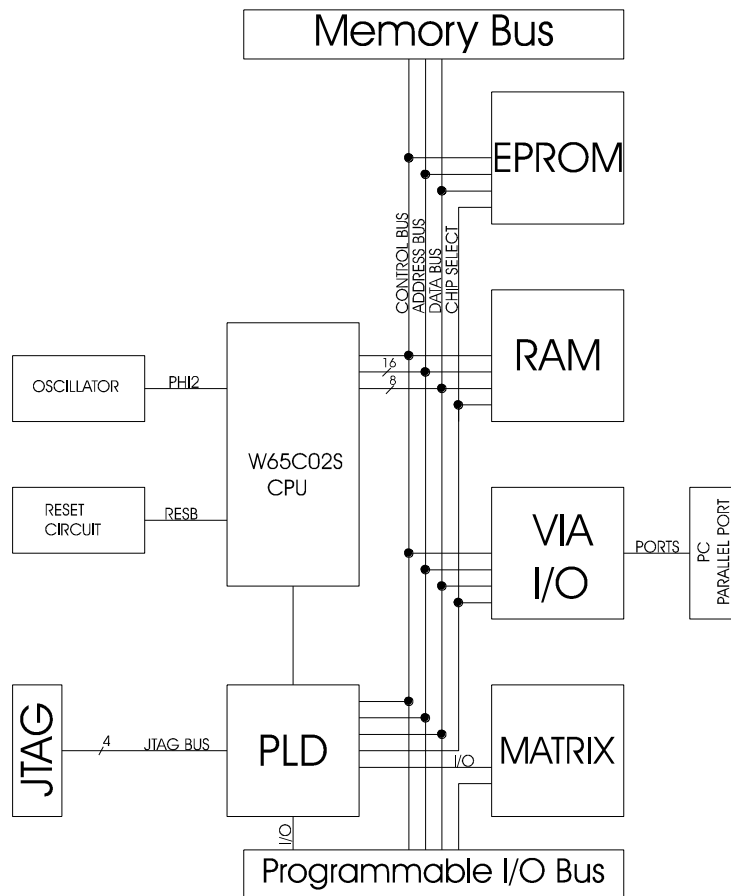


Figure 1.2 W65C02 Developer Board

WDC's developer boards consist of a W65C02S 8-bit MPU or W65C134 8-bit MCU, total access to all control lines, Memory Bus, Programmable I/O Bus, PC Interface, 20 I/O lines, easy oscillator change, 32K SRAM, 32K EPROM, W65C22S Versatile Interface Adapter VIA peripheral chip, on-board matrix, PLD for Memory map decoding and ASIC design. The PLD chip is a XILINX XC9572 for changing the chip select and I/O functions if required. To change the PLD chip to suit your own setup, you need XILINX Data Manager for the XC9572 CPLD chip. The DB includes an on-board programming header for JTAG configuration. For more details refer to the circuit diagram. The on-board W65C02S/W65C134 and the W65C22S devices have measurement points for core power consumption. Power input is provided by an optional power board which plugs into the 10 pin power header. If voltage is required between 1.2-6V, use an external power supply. The IO daughter board can be used to connect the DB to the parallel port of a PC.

#### Memory map:

CS1B:	8000-FFFF	⇒	EPROM (27C256)
CS3B:	0000-00EF & 0100-7FFF	⇒	SRAM (62C256)
CS2B:	00F0-00FF	⇒	VIA (W65C22S)

The new chip to be designed is the W65C122S. The chip has many applications such as toys, industrial controllers, robots, communications, etc. The first thing is to define the pin-out. The pin-out, sometimes called the "footprint" of the W65C134S is shown in Figure 1.3. The footprint of the W65C122S is shown in Figure 1.4.

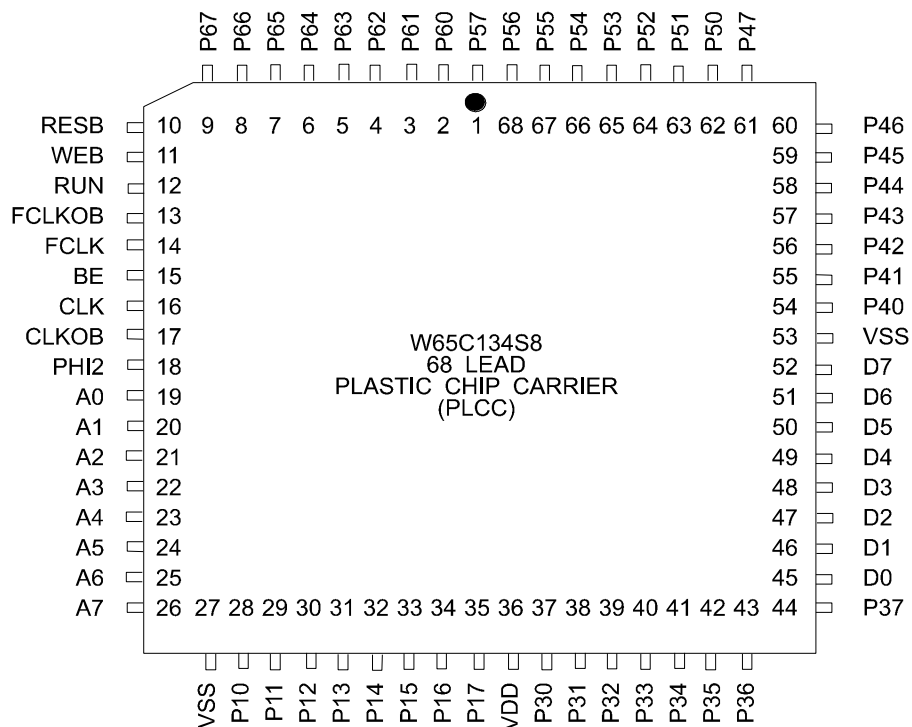


Figure 1.3 W65C134S Footprint

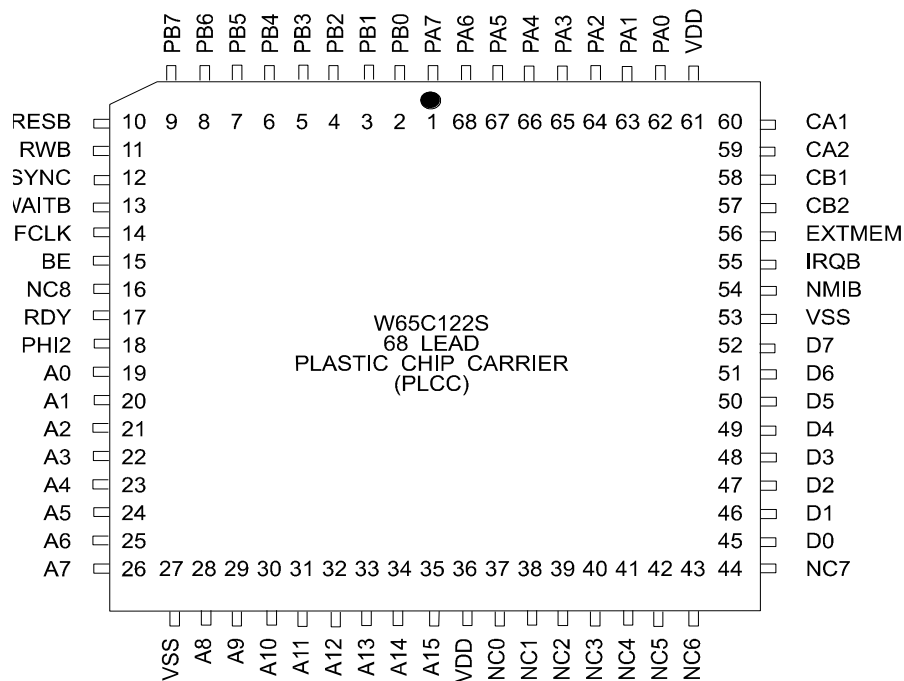


Figure 1.4 W65C122S Footprint

## Chapter 2

### Specification of System-On-a-Chip

#### 2.1 Specification for W65C122S

The W65C122S is a fully static 8-bit SOC microcomputer using a CMOS process. The W65C02 core microprocessor Processor Status Register (P) is shown in Figure 2.1 with the programming model shown in Figure 2.2. The Block Diagram of the W65C122S is given in Figure 2.3.

The W65C122S includes the W65C22S for programmed control of up to two peripheral devices (Ports A and B). Two program controlled 8-bit bi-directional peripheral I/O ports provide direct interfacing between the microprocessor and selected peripheral units. Each port has input data latching capability. Two programmable Data Direction Registers (A and B) allow selection of data direction (input or output) on an individual line basis. Also provided are two programmable 16-bit Interval Timer/Counters with latches. Timer 1 may be operated on a One-Shot Interrupt Mode with Interrupts on each count-to-zero, or in a Free-Run Mode with a continuous series of evenly spaced interrupts. Timer 2 functions as both an interval and pulse counter. Serial Data transfers are provided by a serial-to-parallel/parallel-to-serial shift register. Application versatility is further increased by various control registers, including an Interrupt Flag Register, an Interrupt Enable Register and two Function Control Registers. Refer to the W65C02S and W65C22S Data Sheets for detailed information.

MPU Core	W65C02S
ROM	4096 bytes
RAM	256 bytes
I/O Core	W65C22S
Timer/Counters	2
Synchronous Serial Port	1
Low Power Modes	WAI, STP
Power supply	2.8V-5.5V
Package type	68 PLCC

Table 2.1 Features of the W65C122S

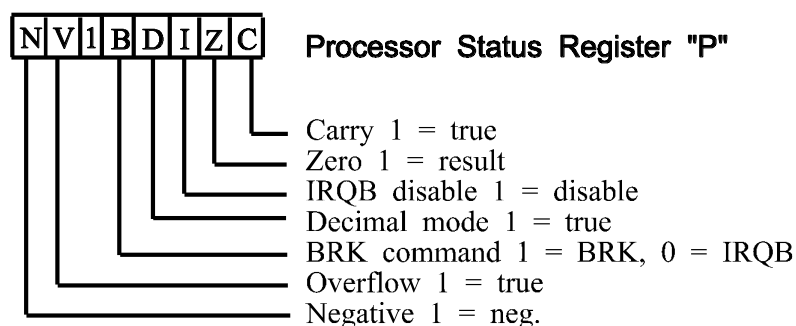


Figure 2.1 W65C02S Status Register Coding



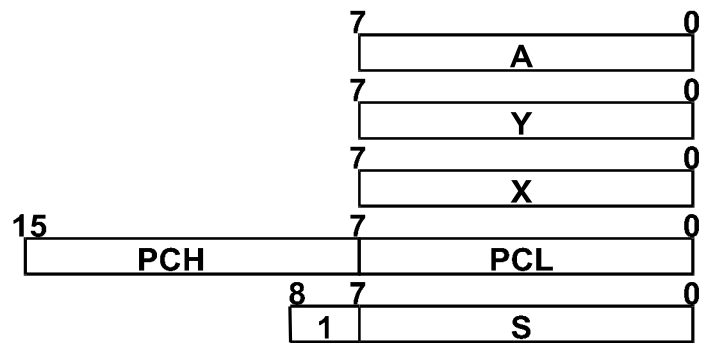


Figure 2.2 W65C02S Programming Model

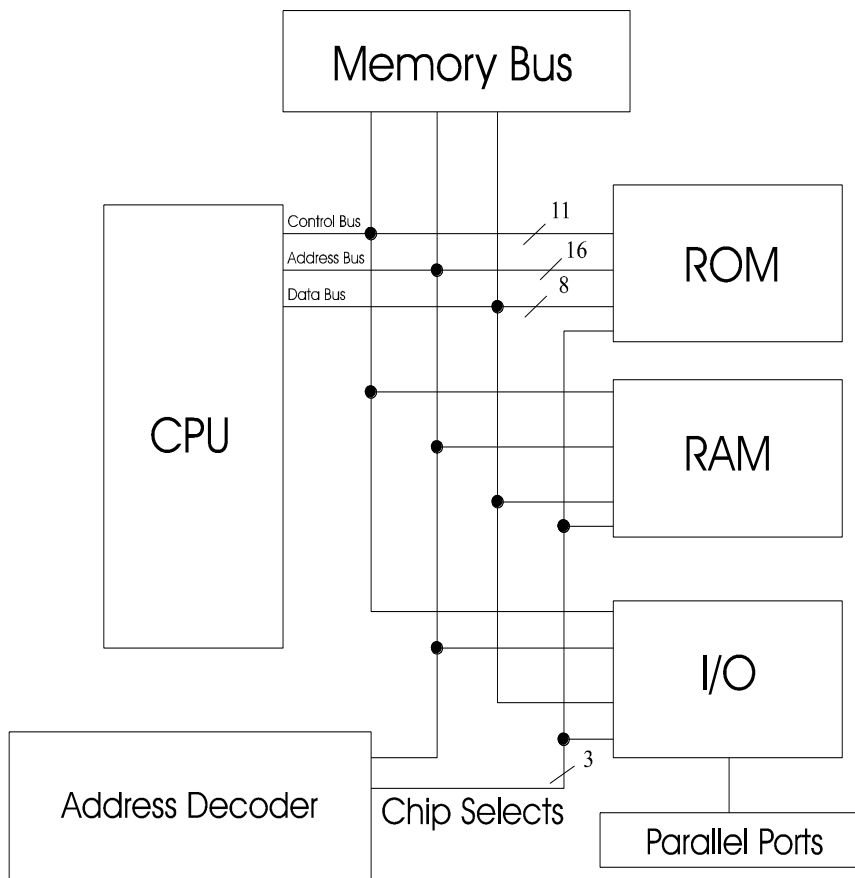


Figure 2.3 Block Diagram of W65C122S

Address	Label	Function
FFFE-FFFF	IRQ	IRQB vector
FFFC-FFFD	RESET	RESETB vector
FFFA-FFFB	NMI	NMIB vector
F000-FFF9	ROM	On chip ROM
0100-01EF	RAM	On chip RAM (Page 1) Stack
00F0-00FF	VIA	On chip VIA
0000-00EF	RAM	On chip RAM (Page 0)

Table 2.2 System Memory Map

## 2.2 Pin descriptions

### 2.2.1 Address Bus(A0- A15)

The 16 –bit Address Bus (A0- A15)is used to address memory and I/O.

### 2.2.2 Bus Enable (BE)

The Bus Enable (BE) input signal controls the address, data and the RWB buffers. When BE enable is high, the RWB, data and address buffers are active. When BE is low, these buffers are in the high impedance state and they may be driven by external circuitry such as Direct Memory Access (DMA) Circuitry. This is an asynchronous signal.

### 2.2.3 Control Lines (CA1, CA2)

The Control Lines (CA1, CA2) serve as the interrupt inputs or handshake outputs for Port A. Each line controls an internal interrupt flag with a corresponding interrupt enable bit. CA1 also controls the latching of Input Data on Port A.(See W65C22S Data Sheet)

### 2.2.4 Control Lines (CB1, CB2)

The Control Lines (CB1, CB2) serve as interrupt inputs or handshake outputs for Port B. These two control lines control an internal interrupt flag with an corresponding interrupt enable bit. These lines also can be configured as a serial data port under control of Shift Register. (See W65C22S Data Sheet)

### 2.2.5 Data Bus (D0-D7)

The Data Bus (D0-D7) is used for data exchanges between microprocessors, memory and peripherals.

### 2.2.6 External Memory (EXTMEMB)

The External Memory (EXTMEMB) input controls the memory map, when EXTMEMB is high the on-chip RAM and ROM are accessed at the memory locations in Table 2.2. When EXTMEMB is low then External memory is accessed at these addresses.

## **2.2.7 FCLK**

This is the main clock source for the system and is driven from an oscillator.

## **2.2.8 Interrupt Request (IRQB)**

The Interrupt Request (IRQB) signal requests that an interrupt sequence begin within the microprocessor. The IRQB is sampled at the falling edge of PHI2 operation. If the Interrupt disable flag (I) in the processor status register is zero, the current instruction is completed and the interrupt sequence begins during PHI2 low. The IRQB is a level sensitive interrupt and therefore must remain in the active low state when RDY is low and IRQB is recognized when RDY goes high and the current instruction is completed. The following sequence occurs:

- The microprocessor stores the program counter and the status register to the stack.
- The microprocessor sets the Interrupt disable flag high (I) so that no further interrupts are recognized
- The Program Counter (PC) is loaded from location FFFE and FFFF. This is the start of the interrupt handler routine. (See W65C02S Data Sheet)

## **2.2.9 No Connect (NC)**

The No Connect (NC) pins are not connected internally and may be connected externally as will occur in the W65C134DB Developer Board for testing. These pins may be used for new features such as On-Chip-Debug (OCD). (See W65C134S Data Sheet)

## **2.2.10 Non-maskable Interrupt (NMIB)**

The Non-maskable Interrupt (NMIB) input is an edge sensitive interrupt that is sampled at the falling edge of PHI2. After the current instruction is completed, the interrupt sequence begins. The program counter is loaded with the vector from FFFA and FFFB (high byte). Since this is a non maskable interrupt, another interrupt will occur while the microprocessor is servicing one. No interrupt can occur if NMIB is low and a negative going edge has not occurred since the last non-maskable interrupt. (See W65C02S Data Sheet)

## **2.2.11 Peripheral Data Port A (PA0-PA7)**

The Peripheral Data Port A (PA0-PA7) is an 8-bit bi-directional bus used for data transfer of data, control, and status information between the W65C122S and a peripheral device. The input data maybe latched into register A of the VIA using the CA1 pin. (See W65C22S Data Sheet)

## **2.2.12 Peripheral Data Port B (PB0- PB7)**

The Peripheral Data Port B (PB0- PB7) is an 8-bit bi-directional bus. The output signal on the PB7 may be controlled by Timer 1 while Timer 2 may be programmed to count pulses on PB6. (See the W65C22S Data Sheet)

## **2.2.13 PHI2**

The PHI2 clock signal is generated from FCLK and provides the timing for the system.

### **2.2.14 Power Supply (VDD, VSS)**

The Power Supply (VDD, VSS) are the positive (VDD) and ground (VSS) power pins. The W65C122S has 2 VDD pins, one for the core, Pin 36 and one for the pad ring, Pin 61. Likewise, there are 2 VSS pins, one for the core, Pin 53 and one for the pad ring, Pin 27. The core and buffer ring power dissipation can be measured independently.

### **2.2.15 Read/Write (RWB)**

The Read/Write (RWB) signal indicates that the microprocessor is reading or writing from memory or I/O bus. When in the high state the microprocessor is “reading” from memory or I/O. In low state, the microprocessor is writing to the addressed memory location. (See W65C02S Data Sheet)

### **2.2.16 Ready (RDY)**

When the Ready (RDY) signal is pulled low, the processor will stop in its current state and will remain in the state until the RDY line goes high.

### **2.2.17 Reset (RESB)**

The Reset (RESB) signal resets the microprocessor stopping all operation and resets the VIA. (See W65C02S and W65C22S Data Sheets) RESB should be held low for at least two cycles after VDD reaches operation voltage from a power down. A positive transition of the signal returns the microprocessor to full operation. The interrupt flag is set, the decimal mode is cleared and the program counter is loaded with the vector from locations FFFC and FFFD (See W65C02S Data Sheet)

### **2.2.18 Synchronize (SYNC)**

The Synchronize (SYNC) signal is high when the microprocessor is fetching an opcode. (See W65C02S Data Sheet)

## WDC STANDARD CELL LIBRARY FOR MASK DESIGN

NAME	DESCRIPTION	
<u>CORES</u>		F:\WDC_LIB\CORE\GDSII\MASK
AB256RAM	W256RAM <sub>TM</sub> core (256 X 8) abstract cell	
AB4KROM	W4KROM <sub>TM</sub> core (4096 X 8) abstract cell	
AB65C02	W65C02C <sub>TM</sub> core abstract cell	
AB65C22	W65C22C <sub>TM</sub> core abstract cell	
<u>STANDARD CELLS</u>		F:\WDC_LIB\STDCELL\GDSII\MASK\CELL
INVS	Small size (20/8) INVerter	
INVM	Medium size (50/20) INVerter	
INVL	Large size (100/40) INVerter	
NA2S	Small size (20/16) 2-input NAND gate	
NA2M	Medium size (50/40) 2-input NAND gate	
NA2B	2-input NAND gate for buffer (100/20)	
NA3S	Small size (20/24) 3-input NAND gate	
NA4S	Small size (20/32) 4-input NAND gate	
NO2S	Small size (40/8) 2-input NOR gate	
NO2M	Medium size (100/20) 2-input NOR gate	
NO2B	2-input NOR gate for buffer (50/40)	
NO3S	Small size (60/8) 3-input NOR gate	
NO4S	Small size (80/8) 4-input NOR gate	
SMS8	Schmitt Trigger	
<u>STANDARD MODULES</u>		F:\WDC_LIB\STDMODUL\GDSII\MASK
IODM	Input/Output Driver Module	
BHD	Bus Holding Device (core)	
OBDM	Output Buffer Driver Module	
OCBL	On Chip Buffer Large – N transistor (40/2), P transistor (100/2)	
<u>STANDARD PADS</u>		F:\WDC_LIB\STDMODUL\GDSII\MASK
IOPD	Input/Output Pad Module (with holding device)	
IOPN	Input/Output Pad Module (without holding device)	
VDDC	Core VDD power pad cell	
VDDP	Pad ring VDD power pad cell	
VSSC	Core VSS power pad cell	
VSSP	Pad ring VSS power pad cell	
CORNER	Corner cell (mask design only)	
SPACER	Spacer cell (mask design only)	
<u>APPLICATION SPECIFIC MODULES</u>		F:\WDC_LIB\STDMODUL\GDSII\MASK\ADDECODE
ADDECODE	Address Decoder	

\*Note: The “ELEMENTS” directory exists only for schematic data – not for mask data

Table 2.3 WDC Standard Cell Library for Mask Design

## WDC STANDARD CELL LIBRARY FOR SCHEMATIC DESIGN

NAME	DESCRIPTION		
<u>CORES</u>			
AB256RAM	W256RAM™ core (256 X 8) abstract cell	F:\WDC_LIB\CORE\GDSII\SCHEMATI	
AB4KROM	W4KROM™ core (4096 X 8) abstract cell		
AB65C02	W65C02C™ core abstract cell		
AB65C22	W65C22C™ core abstract cell		
<u>STANDARD CELLS</u>			
INVS	Small size (20/8) INVerter	F:\WDC_LIB\STDCELL\GDSII\SCHEMATI\CELL	
INVM	Medium size (50/20) INVerter		
INVL	Large size (100/40) INVerter		
NA2S	Small size (20/16) 2-input NAND gate		
NA2M	Medium size (50/40) 2-input NAND gate		
NA2B	2-input NAND gate for buffer (100/20)		
NA3S	Small size (20/24) 3-input NAND gate		
NA4S	Small size (20/32) 4-input NAND gate		
NO2S	Small size (40/8) 2-input NOR gate		
NO2M	Medium size (100/20) 2-input NOR gate		
NO2B	2-input NOR gate for buffer (50/40)		
NO3S	Small size (60/8) 3-input NOR gate		
NO4S	Small size (80/8) 4-input NOR gate		
SM28	Schmitt Trigger		
<u>ELEMENTS</u>			
INVERT	INVERTER shape only *	F:\WDC_LIB\STDCELL\GDSII\SCHEMATI\ELEMENT	
NAND	NAND shape only *		
NOR	NOR shape only *		
BUBBLE	Schematic bubble shape only *		
CUPL	Schematic couple shape only *		*Elements used to create standard cells and schematics. Not used for mask data.
DOT	Schematic dot shape only *		
PLAGATE	Schematic shape for gates in PLA *		
VDD	Schematic VDD symbol only *		
VSS	Schematic VSS symbol only *		
SCHVSSPD	Schematic symbol for ground to VSS *		
SCHVDDPD	Schematic symbol for ground to VDD *		
SCHNC	Schematic symbol for No/Connect *		
<u>STANDARD MODULES</u>			
IODM	Input/Output Driver Module	F:\WDC_LIB\STDMODUL\GDSII\SCHEMATI	
BHD	Bus Holding Device (core)		
OBDM	Output Buffer Driver Module		
OCBL	On Chip Buffer Large – N transistor (40/2), P transistor (100/2)		
<u>STANDARD PADS</u>			
IOPD	Input/Output Pad Module (with holding device)	F:\WDC_LIB\STDMODUL\GDSII\SCHEMATI	
IOPN	Input/Output Pad Module (without holding device)		
VDDC	Core VDD power pad cell		
VDDP	Pad ring VDD power pad cell		
VSSC	Core VSS power pad cell		
VSSP	Pad ring VSS power pad cell		
<u>APPLICATION SPECIFIC MODULES</u>			
ADDECODE	Address Decoder	F:\WDC_LIB\STDMODUL\GDSII\SCHEMATI\ADDECODE	

\*Elements used to create standard cells and schematics. Not used for mask data.

Table 2.4 WDC Standard Cell Library for Schematic Design







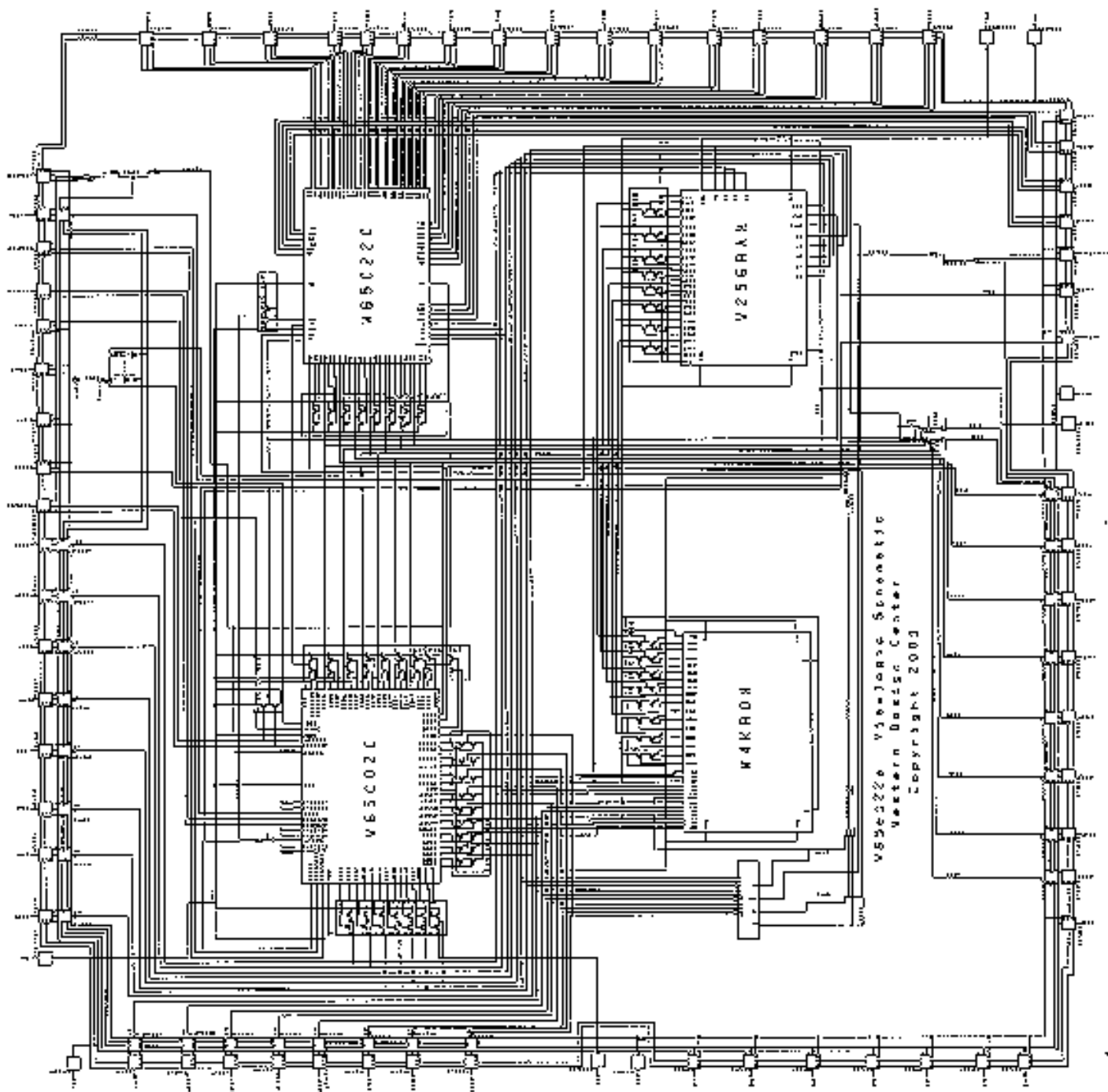
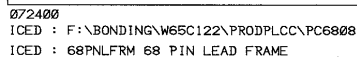
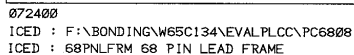


Figure 2.8 W65C122S Viewlogic Schematic






## Chapter 3

### Design Methodology

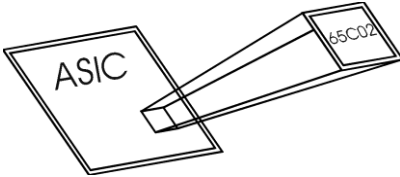
#### 3.1 Design Tools

Figure 3.1 shows the EDA design tools most commonly used in the industry. The design tools required for this SOC design are:

1. Recommended PC configuration: Pentium, 64MB or more RAM, 2GB or larger hard disk and 200MHz or faster speed
2. Windows 95
3. Excel spreadsheet software required for GDSII design rules
4. Text editor
5. GDSII editor, ICED32 preferred for ROM placement using WDC utility
6. PSpice simulator
7. Verilog Silos III simulator required for encrypted models
8. Viewlogic software



### ASIC Design Tools



Design Task	Tool Name	Tool Vendor	Industry Standard Format
Design Entry	Verilog	Text editor	Verilog, Structural (gate), RTL, Behavioral
Simulation	Silos III	Simucad	Verilog
Synthesis	Leonardo	Exemplar	Verilog/EDIF
Graphic Design	ICED32	IC Editors	GDSII
Timing Analysis	PSpice	MicroSim	Spice
Automated Test	Sentry	Schlumberger	Factor/TRASCII

Figure 3.1 Design Tools

## 3.2 Design Flow for Retargeting

The design flow for retargeting a hard core and an RTL model require different steps. A firm core is defined as a synthesized and placed and routed core of the RTL model. A GDSII firm core is defined as a synthesized and placed and routed core of the RTL model using the GDSII standard cell library.

Figure 3.2 shows the design flow for both the hard core and an RTL model.

The optimized GDSII hard core steps include: 1) select the desired process, 2) retarget the GDSII layers, 3) extract the Spice netlist, 4) simulate for new timing, and 5) update the timing model. The core is ready to embed in the desired application. Not only are there half as many steps required with the hard core, there is also significantly less effort per step since the GDSII data base was originally proven in silicon. With the block place and route system for system-on-chip design, the hard core is thought of as a standard cell and the design flow follows the standard cell retargeting flow.

The design flow to retarget the RTL model is more complicated. The RTL model steps include: 1) select the process, 2) select the library, 3) retarget the library if it hasn't been retargeted 4) synthesize with the library, 5) floor plan for placing and routing, 6) place and route the GDSII standard cells, 7) extract a netlist for verification, 8) simulate to ensure the desired timing is achieved, and 9) update the timing model. If, at this time, the firm core doesn't meet the desired criteria, then repeat steps 4-9. The GDSII firm core is now ready to embed.

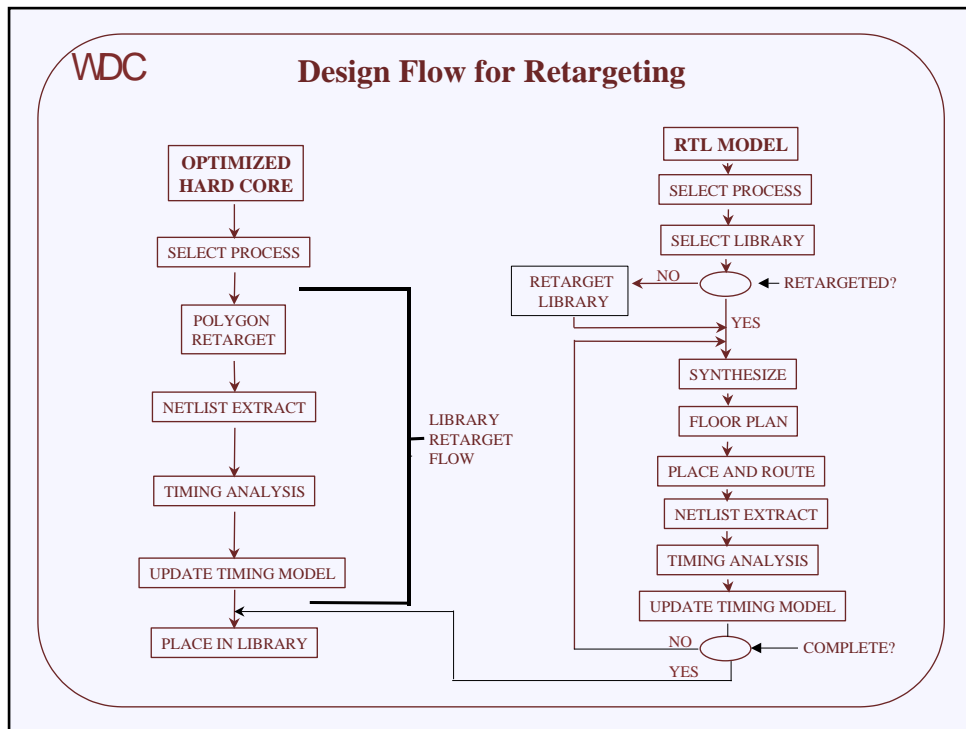


Figure 3.2 Design Flow for Retargeting

### 3.2.1 Retargeting flow for MOSIS manufacturing

1. Select the targeted MOSIS process
2. Enter target rules into WDC Retargetable Design Rules Excel spreadsheet for cores
3. Convert layers from WDC layers to the targeted process layers
4. Scale core using scale factor from Excel spreadsheet for cores
5. Bias using bias calculations from Excel spreadsheet
6. DRC core using the targeted process core design rules
7. Enter target rules in WDC Retargetable Design Rules Excel spreadsheet for pad ring
8. Scale ring using scale factor from Excel spreadsheet for pad ring
9. DRC pad ring using the targeted process design rules for I/O
10. Wire the core to the ring
11. DRC using the targeted process design rules
12. Extract Spice netlist
13. LVS using extracted netlist and W65C122S<sup>TM</sup> CDL netlist
14. Perform Spice simulation in Spice test bench with extracted Spice netlist
15. Create scribe ring and add to chip
16. Ship completed design to MOSIS indicating use of native vendor design rules
17. Contract MOSIS to DRC design using the targeted process design rules (optional)
18. Contract MOSIS to build chips
19. Receive chips from MOSIS
20. Test chip in WDC's W65C134 Developer Board

# GLOSSARY

## **Behavioral Model**

A description of the function and timing of a component without describing a specific implementation. A behavioral model can exist at any level of abstraction. Abstraction depends on the precision of implementation details. For example, a behavioral model can be a model that describes the bulk time and functionality of a processor that executes an abstract algorithm, or it can be a model of the processor at the less abstract instructions set level. The precision of internal and external data values depends on the model's abstraction level.

## **Black Box**

An implementation of a hard, firm, or soft VC that is hidden from the designer. It is only observable as a bus functional model at the I/O ports.

## **Firm VC**

VCs that been structurally and topologically optimized for performance and area through floorplanning and/or placement using a generic technology library. The level of detail ranges from region placement of RTL sub-blocks, to relatively placed datapaths, to parametrized generators, to a fully placed netlist. A combination of these approaches is often used to meet the design goals. Firm VCs offer a compromise between soft VCs and hard VCs. They are more flexible and portable than hard VCs, yet more predictive of performance and area than soft VCs. Firm VCs include a combination of synthesizable RTL, reference technology library, detailed floorplan, and a full or partial netlist. When a full netlist is present, it is expected that the test logic has been inserted and that the test lists will accompany the design. Firm VCs do not include routing. Protection risk is equivalent to soft.

## **Hard VC**

VCs that have been optimized for power, size, or performance, and mapped to a specific technology. Examples include netlists fully placed, routed, and optimized for a specific technology library, a custom physical layout, or a combination of the two. Hard VCs are process/vendor specific and generally expressed in GDSII format. They have the advantage of being much more predictable, but consequently are less flexible and portable due to process dependencies. Hard VCs require, at a minimum, a high-level behavioral model, a test list, full physical and timing models along with GDSII data. The ability to legally protect hard VCs is much better because there is no requirement for RTL.

## **Intellectual Property**

Intellectual Property includes, but is not limited to, patents, copyrights, maskworks, trademarks and trade secrets.

## **Physical Blocks**

A model of the physical implementation of the VC and the system chip.

## **Register Transfer Level (RTL) Model**

An RTL model describes a system in terms of registers, combinational circuitry, low-level buses, and control circuits, usually implemented as finite state machines. Some internal structural implementation formation is implied by the register transformations, but this information is not explicitly described. The primary purpose of RTL models is for developing and testing the internal architecture and control logic with an IC component so the design satisfies the required functionality and timing constraints of the IC. The RTL model is also used for specifying the design in a process neutral format that is retargetable to specific technologies or process lines through automatic synthesis. It is often used for generating detailed test vectors, gathering timing measurements to increase the accuracy of more abstract models, investigating interactions with loosely connected components, and it unambiguously documents the design solution.

## **RTL Source**

Defines the VC source description and is the primary input for the implementation and verification of the VC within a system chip design.

<b>Soft VC</b>	VCs that are delivered in the form of synthesizable HDL code. The advantage is the flexibility of the source code so it can be retargeted to multiple manufacturing processes. The disadvantage is the difficulty in performance prediction (such as timing, area, power). Soft VCs typically have higher intellectual property protection risks because RTL source code is required by the integrator.
<b>Structural Model</b>	A representation of a component or system in terms of the interconnections of its constituent components. A structural model follows the physical hierarchy of the system. The hierarchy reflects the physical organization of a specific implementation. A structural model describes the physical structure of a specific implementation by specifying the components and their topological interconnections. These components can be described structurally, functionally, or behaviorally. Simulation of a structural model requires all models in the lowest (leaf) branches of the hierarchy to be behavioral or functional models. Therefore, the effective temporal, data value, and functional precisions depend on the leaf models. A structural model can exist at any level of abstraction. Structural precision depends on the granularity of the structural blocks.
<b>System Chip</b>	A term used to describe highly integrated devices. It is also known as system on silicon, system-on-a-chip, system-LSI, system-ASIC, and a System-Level-Integration (SLI) device.
<b>Testbench</b>	Test system environment in which the core is tested. Includes the generators for the various control signals and the system memory and interface. Also includes simulation data output.
<b>Virtual Component (VC)</b>	A block that meets the VSIA specification and is used as a component in the virtual socket design environment. Virtual Components can be of the forms: soft, firm, or hard. A pre-implemented, reusable module of intellectual property that can be quickly inserted and verified to create a single-chip system. VCs are also called megacells or cores.