Chipsmall Limited consists of a professional team with an average of over 10 year of expertise in the distribution of electronic components. Based in Hongkong, we have already established firm and mutual-benefit business relationships with customers from,Europe,America and south Asia,supplying obsolete and hard-to-find components to meet their specific needs.

With the principle of “Quality Parts,Customers Priority,Honest Operation,and Considerate Service”,our business mainly focus on the distribution of electronic components. Line cards we deal with include Microchip,ALPS,ROHM,Xilinx,Pulse,ON,Everlight and Freescale. Main products comprise IC,Modules,Potentiometer,IC Socket,Relay,Connector.Our parts cover such applications as commercial,industrial, and automotives areas.

We are looking forward to setting up business relationship with you and hope to provide you with the best service and solution. Let us make a better world for our industry!

Contact us

Tel: +86-755-8981 8866 Fax: +86-755-8427 6832
Email & Skype: info@chipsmall.com Web: www.chipsmall.com
Address: A1208, Overseas Decoration Building, #122 Zhenhua RD., Futian, Shenzhen, China

Features

- Monolithic Field Programmable System Level Integrated Circuit (FPSLIC™)
  - AT40K SRAM-based FPGA with Embedded High-performance RISC AVR® Core, Extensive Data and Instruction SRAM and JTAG ICE
- 5,000 to 40,000 Gates of Patented SRAM-based AT40K FPGA with FreeRAM™
  - 2 - 18.4 Kbits of Distributed Single/Dual Port FPGA User SRAM
  - High-performance DSP Optimized FPGA Core Cell
  - Dynamically Reconfigurable In-System – FPGA Configuration Access Available On-chip from AVR Microcontroller Core to Support Cache Logic® Designs
  - Very Low Static and Dynamic Power Consumption – Ideal for Portable and Handheld Applications
- Patented AVR Enhanced RISC Architecture
  - 120+ Powerful Instructions – Most Single Clock Cycle Execution
  - High-performance Hardware Multiplier for DSP-based Systems
  - Approaching 1 MIPS per MHz Performance
  - C Code Optimized Architecture with 32 x 8 General-purpose Internal Registers
  - Low-power Idle, Power-save and Power-down Modes
  - 100 µA Standby and Typical 2-3 mA per MHz Active
- Up to 36 Kbytes of Dynamically Allocated Instruction and Data SRAM
  - Up to 16 Kbytes x 16 Internal 15 ns Instructions SRAM
  - Up to 16 Kbytes x 8 Internal 15 ns Data SRAM
- JTAG (IEEE std. 1149.1 Compliant) Interface
  - Extensive On-chip Debug Support
  - Limited Boundary-scan Capabilities According to the JTAG Standard (AVR Ports)
- AVR Fixed Peripherals
  - Industry-standard 2-wire Serial Interface
  - Two Programmable Serial UARTs
  - Two 8-bit Timer/Counters with Separate Prescaler and PWM
  - One 16-bit Timer/Counter with Separate Prescaler, Compare, Capture Modes and Dual 8-, 9- or 10-bit PWM
- Support for FPGA Custom Peripherals
  - AVR Peripheral Control – 16 Decoded AVR Address Lines Directly Accessible to FPGA
  - FPGA Macro Library of Custom Peripherals
- 16 FPGA Supplied Internal Interrupts to AVR
- Up to Four External Interrupts to AVR
- 8 Global FPGA Clocks
  - Two FPGA Clocks Driven from AVR Logic
  - FPGA Global Clock Access Available from FPGA Core
- Multiple Oscillator Circuits
  - Programmable Watchdog Timer with On-chip Oscillator
  - Oscillator to AVR Internal Clock Circuit
  - Software-selectable Clock Frequency
  - Oscillator to Timer/Counter for Real-time Clock
- V_{CC}: 3.0V - 3.6V
- 3.3V 33 MHz PCI-compliant FPGA I/O
  - 20 mA Sink/Source High-performance I/O Structures
  - All FPGA I/O Individually Programmable
- High-performance, Low-power 0.35µ CMOS Five-layer Metal Process
- State-of-the-art Integrated PC-based Software Suite including Co-verification
- 5V I/O Tolerant

FPSLIC™

5K - 40K Gates of AT40K FPGA with 8-bit AVR® Microcontroller, up to 36K Bytes of SRAM and On-chip JTAG ICE

AT94KAL Series Field Programmable System Level Integrated Circuit

Rev. 1138G–FPSLI–11/03

Description

The AT94KAL Series FPSLIC family shown in Table 1 is a combination of the popular Atmel AT40K Series SRAM FPGAs and the high-performance Atmel AVR 8-bit RISC microcontroller with standard peripherals. Extensive data and instruction SRAM as well as device control and management logic are included on this monolithic device, fabricated on Atmel's 0.35µ five-layer metal CMOS process.

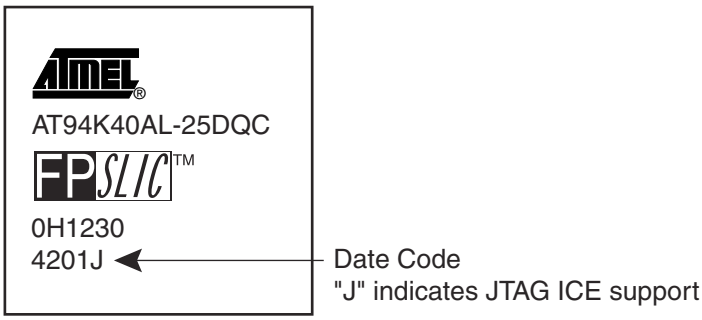
The AT40K FPGA core is a fully 3.3V PCI-compliant, SRAM-based FPGA with distributed 10 ns programmable synchronous/asynchronous, dual-port/single-port SRAM, 8 global clocks, Cache Logic ability (partially or fully reconfigurable without loss of data) and 5,000 to 40,000 usable gates.

Table 1. The AT94K Series Characteristics

Device		AT94K05AL	AT94K10AL	AT94K40AL
FPGA Gates		5K	10K	40K
FPGA Core Cells		256	576	2304
FPGA SRAM Bits		2048	4096	18432
FPGA Registers (Total)		436	846	2862
Maximum FPGA User I/O		96	144	288
AVR Programmable I/O Lines		8	16	16
Program SRAM		4 Kbytes - 16 Kbytes	20 Kbytes - 32 Kbytes	20 Kbytes - 32 Kbytes
Data SRAM		4 Kbytes - 16 Kbytes	4 Kbytes- 16 Kbytes	4 Kbytes - 16 Kbytes
Hardware Multiplier (8-bit)		Yes	Yes	Yes
2-wire Serial Interface		Yes	Yes	Yes
UARTs		2	2	2
Watchdog Timer		Yes	Yes	Yes
Timer/Counters		3	3	3
Real-time Clock		Yes	Yes	Yes
JTAG ICE		Yes[1]	Yes[1]	Yes[1]
Typical AVR throughput	@ 25 MHz	19 MIPS	19 MIPS	19 MIPS
Operating Voltage[2]	AL	3.0 - 3.6V[2]	3.0 - 3.6V[2]	3.0 - 3.6V[2]

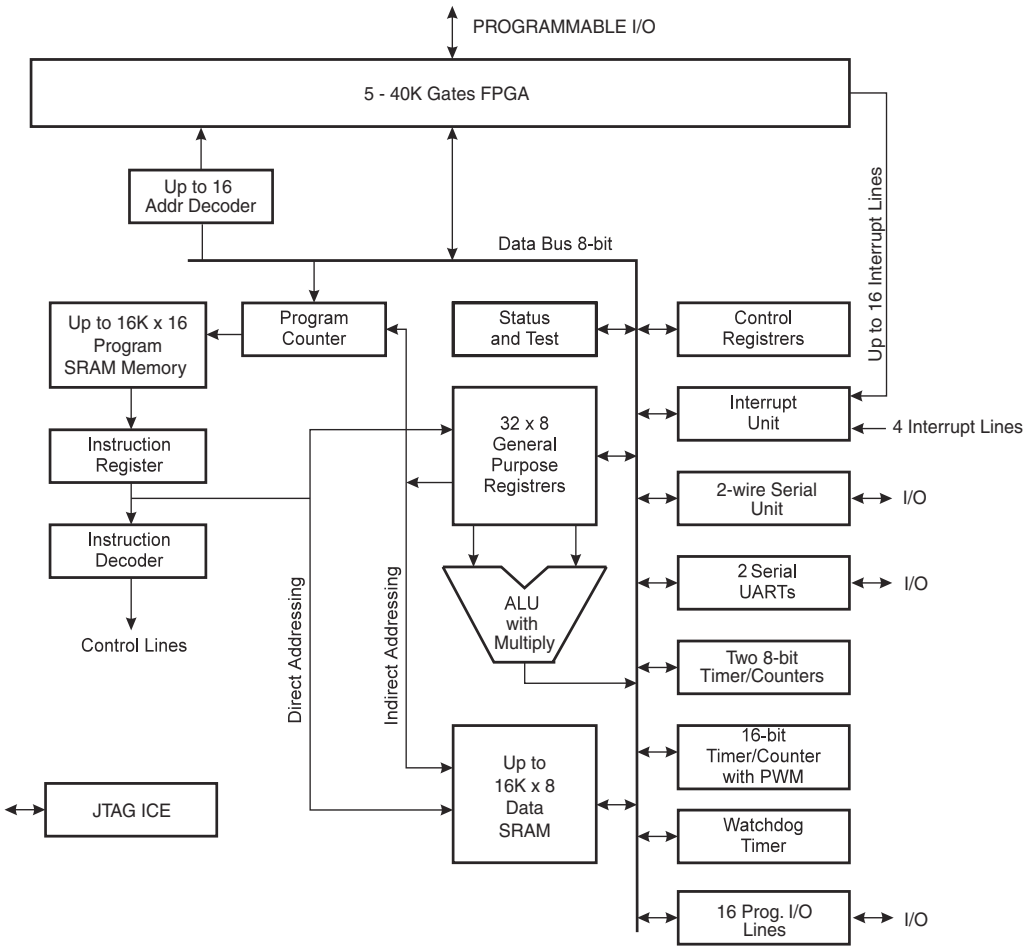
Notes: 1. FPSLIC parts with JTAG ICE support can be identified by the letter "J" after the device date code, e.g., 4201 (no ICE support) and 4201J (with ICE support), see Figure 1.
2. FPSLIC devices should be laid out during PCB design to support a split power supply. Please refer to the "Designing in Split Power Supply Support for AT94KAL and AT94SAL Devices" application note, available on the Atmel web site at http://www.atmel.com/atmel/acrobat/doc2308.pdf.

Figure 1. FPSLIC Device Date Code with JTAG ICE Support



The AT94K series architecture is shown in Figure 2.

Figure 2. AT94K Series Architecture



The embedded AVR core achieves throughputs approaching 1 MIPS per MHz by executing powerful instructions in a single-clock cycle, and allows system designers to optimize power consumption versus processing speed. The AVR core is based on an enhanced RISC architecture that combines a rich instruction set with 32 general-purpose working registers. All 32 registers are directly connected to the Arithmetic Logic Unit (ALU), allowing two independent registers to be accessed in one single instruction executed in one clock cycle. The resulting architecture is more code-efficient while achieving throughputs up to ten times faster than conventional CISC microcontrollers at the same clock frequency. The AVR executes out of on-chip SRAM. Both the FPGA configuration SRAM and the AVR instruction code SRAM can be automatically loaded at system power-up using Atmel's In-System Programmable (ISP) AT17 Series EEPROM Configuration Memories or ATFS FPSLIC Support Devices.

State-of-the-art FPSLIC design tools, System Designer™, were developed in conjunction with the FPSLIC architecture to help reduce overall time-to-market by integrating microcontroller development and debug, FPGA development and Place and Route, and complete system co-verification in one easy-to-use software tool.

Table 2. ATFS FPSLIC Support Devices

FPSLIC Device	FPSLIC Support Device	Configuration Data	Spare Memory
AT94K05	ATFS05	226520 Bits	35624 Bits
AT94K10	ATFS10	430488 Bits	93800 Bits
AT94K40	ATFS40	815382 Bits	233194 Bits

FPGA Core

The AT40K core can be used for high-performance designs, by implementing a variety of compute-intensive arithmetic functions. These include adaptive finite impulse response (FIR) filters, fast Fourier transforms (FFT), convolvers, interpolators, and discrete-cosine transforms (DCT) that are required for video compression and decompression, encryption, convolution and other multimedia applications.

Fast, Flexible and Efficient SRAM

The AT40K core offers a patented distributed 10 ns SRAM capability where the RAM can be used without losing logic resources. Multiple independent, synchronous or asynchronous, dual-port or single-port RAM functions (FIFO, scratch pad, etc.) can be created using Atmel's macro generator tool.

Fast, Efficient Array and Vector Multipliers

The AT40K cores patented 8-sided core cell with direct horizontal, vertical and diagonal cell-to-cell connections implements ultra-fast array multipliers without using any busing resources. The AT40K core's Cache Logic capability enables a large number of design coefficients and variables to be implemented in a very small amount of silicon, enabling vast improvement in system speed.

Cache Logic Design

The AT40K FPGA core is capable of implementing Cache Logic (dynamic full/partial logic reconfiguration, without loss of data, on-the-fly) for building adaptive logic and systems. As new logic functions are required, they can be loaded into the logic cache without losing the data already there or disrupting the operation of the rest of the chip; replacing or complementing the active logic. The AT40K FPGA core can act as a reconfigurable resource within the FPSLIC environment.

Automatic Component Generators

The AT40K is capable of implementing user-defined, automatically generated, macros; speed and functionality are unaffected by the macro orientation or density of the target device. This enables the fastest, most predictable and efficient FPGA design approach and minimizes design risk by reusing already proven functions. The Automatic Component Generators work seamlessly with industry-standard schematic and synthesis tools to create fast, efficient designs.

The patented AT40K architecture employs a symmetrical grid of small yet powerful cells connected to a flexible busing network. Independently controlled clocks and resets govern every column of four cells. The FPSLIC device is surrounded on three sides by programmable I/Os.

Core usable gate counts range from 5,000 to 40,000 gates and 436 to 2,864 registers. Pin locations are consistent throughout the FPSLIC family for easy design migration in the same package footprint.

The Atmel AT40K FPGA core architecture was developed to provide the highest levels of performance, functional density and design flexibility. The cells in the FPGA core array are small, efficient and can implement any pair of Boolean functions of (the same) three inputs or any single Boolean function of four inputs. The cell's small size leads to arrays with large numbers of cells. A simple, high-speed busing network provides fast, efficient communication over medium and long distances.

The Symmetrical Array

At the heart of the Atmel FPSLIC architecture is a symmetrical array of identical cells. The array is continuous from one edge to the other, except for bus repeaters spaced every four cells, see Figure 3. At the intersection of each repeater row and column is a 32 x 4 RAM block accessible by adjacent buses. The RAM can be configured as either a single-ported or dual-ported RAM, with either synchronous or asynchronous operation.

The Busing Network

Figure 3. Busing Network

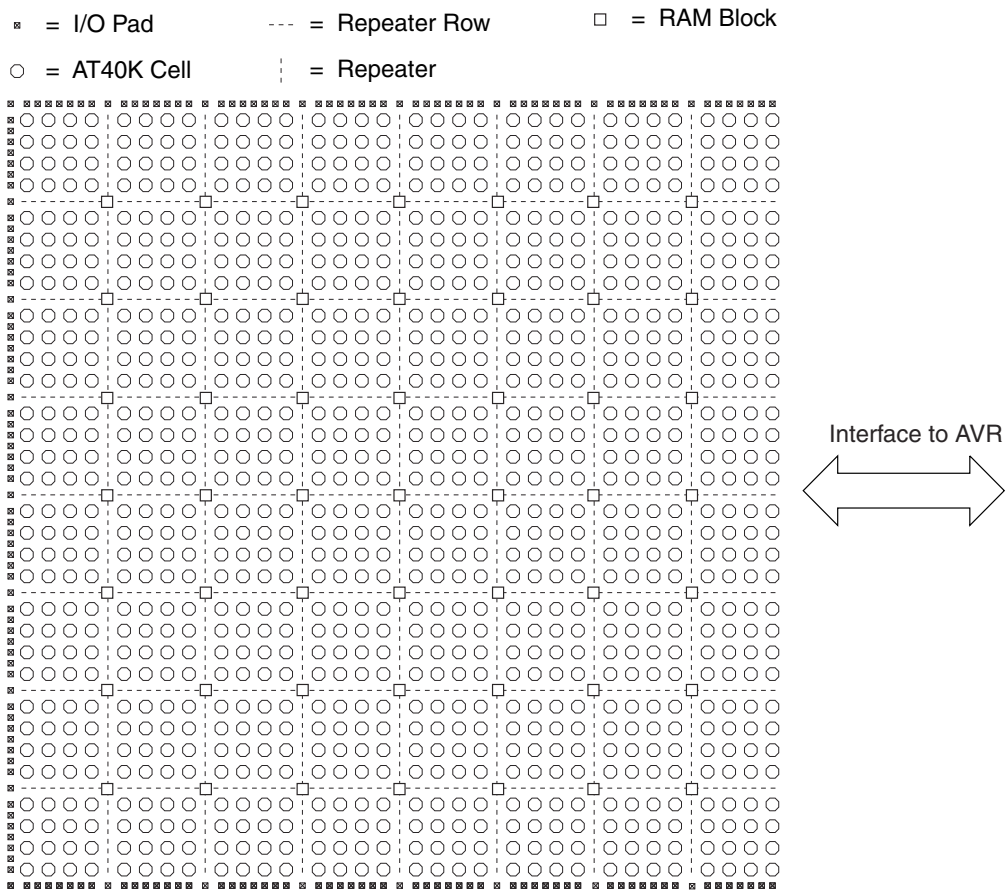
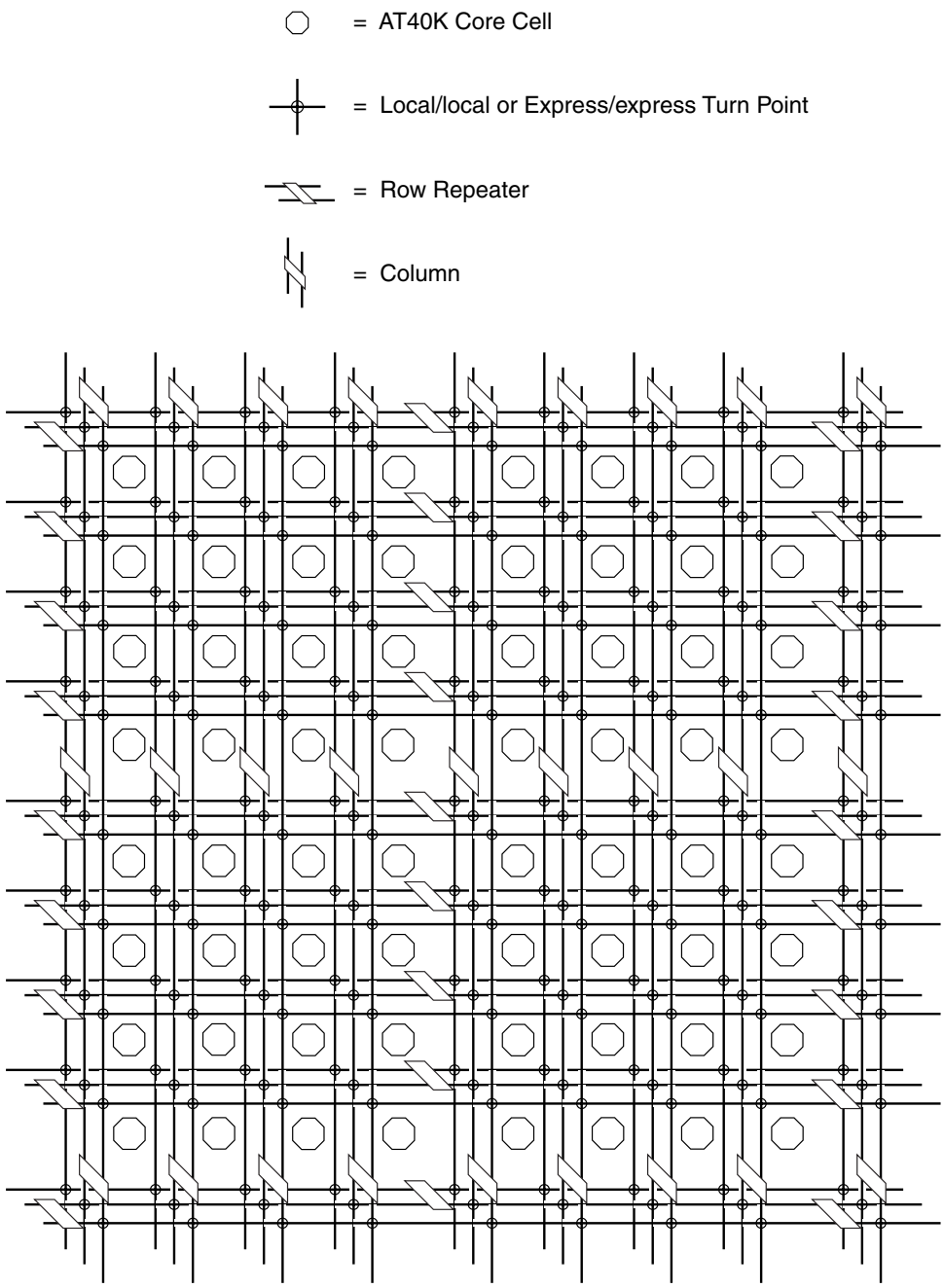


Figure 4 depicts one of five identical FPGA busing planes. Each plane has three bus resources: a local-bus resource (the middle bus) and two express-bus resources. Bus resources are connected via repeaters. Each repeater has connections to two adjacent local-bus segments and two express-bus segments. Each local-bus segment spans four cells and connects to consecutive repeaters. Each express-bus segment spans eight cells and bypasses a repeater. Repeaters regenerate signals and can connect any bus to any other bus (all pathways are legal) on the same plane. Although not shown, a local bus can bypass a repeater via a programmable pass gate, allowing long on-chip tri-state buses to be created. Local/local turns are implemented through pass gates in the cell-bus interface. Express/express turns are implemented through separate pass gates distributed throughout the array.

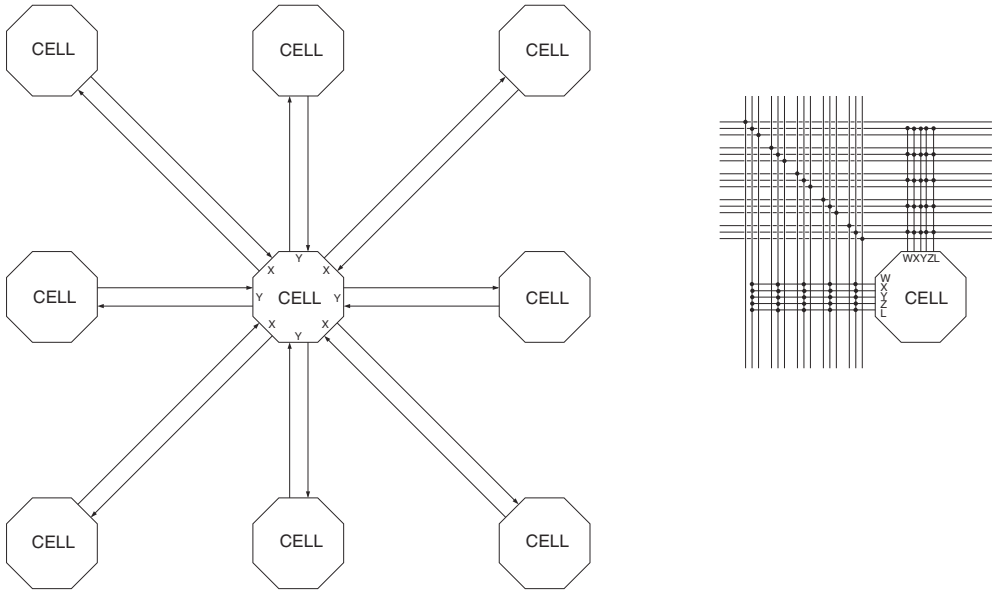
Figure 4. Busing Plane (One of Five)



Cell Connections

Figure 5(a) depicts direct connections between an FPGA cell and its eight nearest neighbors. Figure 5(b) shows the connections between a cell five horizontal local buses (one per busing plane) and five vertical local buses (one per busing plane).

Figure 5. Cell Connections



(a) Cell-to-Cell Connections

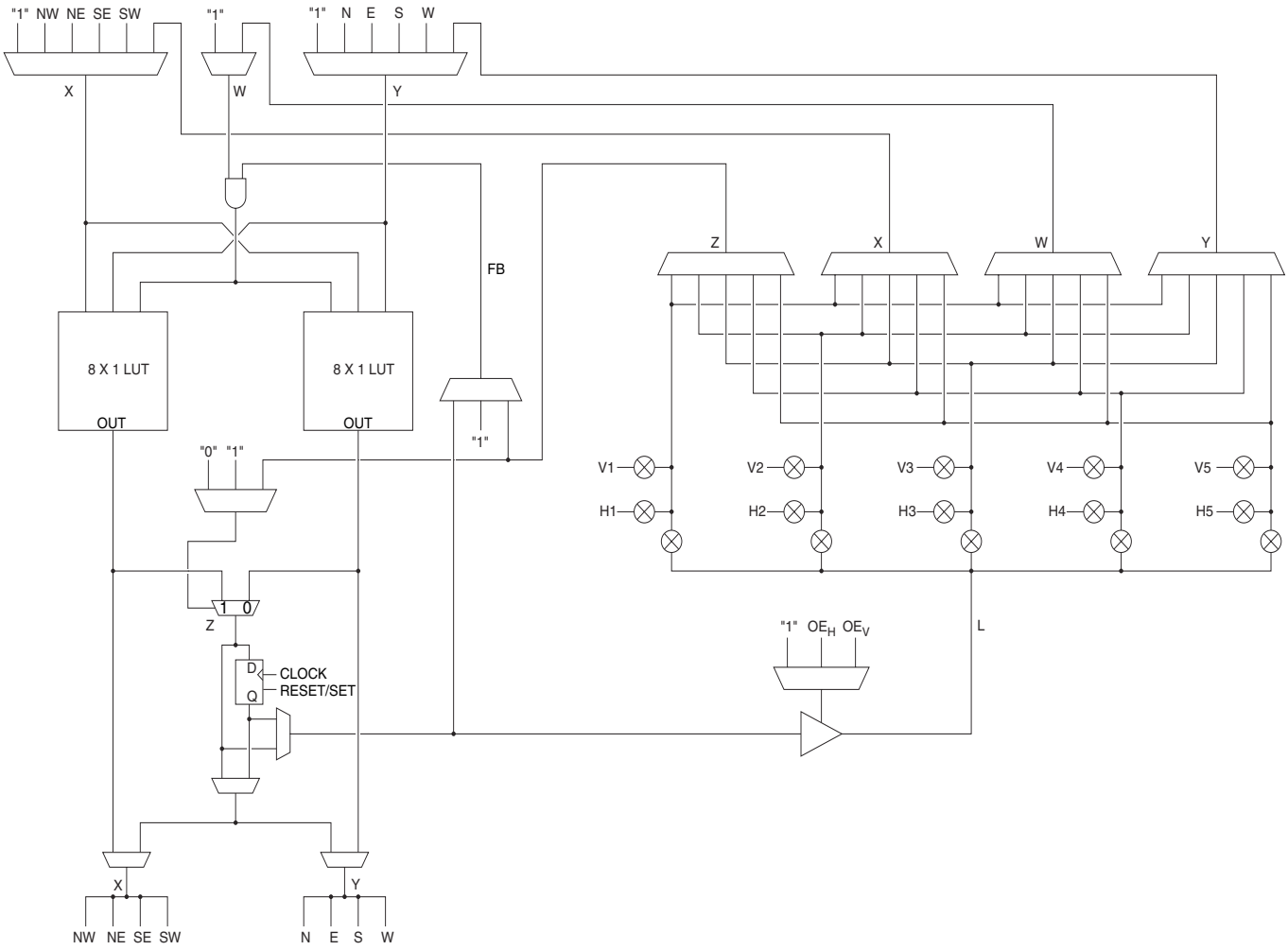
(b) Cell-to-Bus Connections

The Cell

Figure 6 depicts the AT40K FPGA embedded core logic cell. Configuration bits for separate muxes and pass gates are independent. All permutations of programmable muxes and pass gates are legal. Vn is connected to the vertical local bus in plane n. Hn is connected to the horizontal local bus in plane n. A local/local turn in plane n is achieved by turning on the two pass gates connected to Vn and Hn. Up to five simultaneous local/local turns are possible.

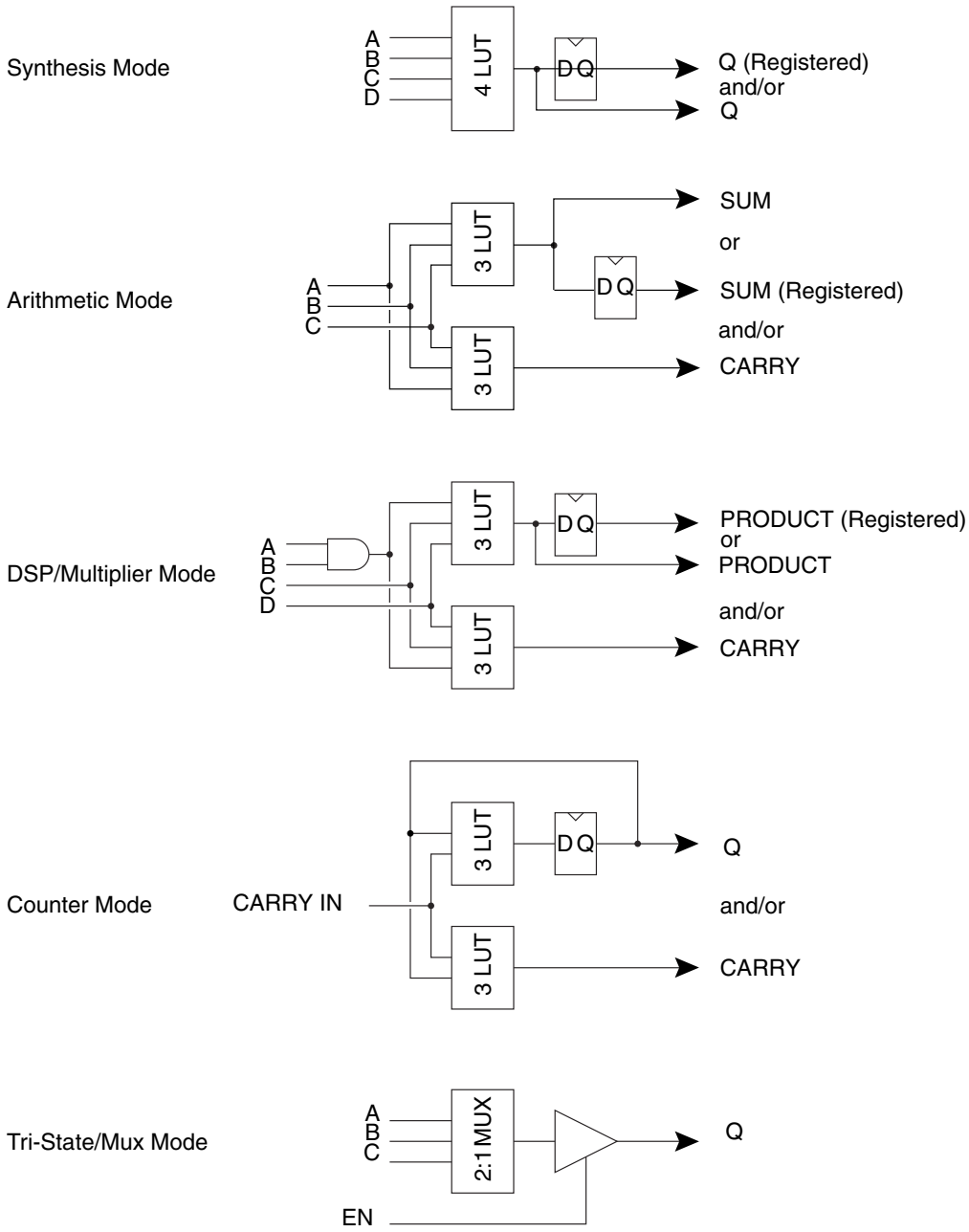
The logic cell can be configured in several “modes”. The logic cell flexibility makes the FPGA architecture well suited to all digital design application areas, see Figure 7. The IDS layout tool automatically optimizes designs to utilize the cell flexibility.

Figure 6. The Cell



X = Diagonal Direct Connect or Bus
Y = Orthogonal Direct Connect or Bus
W = Bus Connection
Z = Bus Connection
FB = Internal Feedback

Figure 7. Some Single Cell Modes



RAM

There are two types of RAM in the FPSLIC device: the FreeRAM distributed through the FPGA Core and the SRAM shared by the AVR and FPGA. The SRAM is described in "FPGA/AVR Interface and System Control" on page 21. The 32 x 4 dual-ported FPGA FreeRAM blocks are dispersed throughout the array and are connected in each sector as shown in Figure 8. A four-bit Input Data bus connects to four horizontal local buses (Plane 1) distributed over four sector rows. A four-bit Output Data bus connects to four horizontal local buses (Plane 2) distributed over four sector rows. A five-bit Input-address bus connects to five vertical express buses in the same sector column (column 3). A five-bit Output-address bus connects to five vertical express buses in the same column. WAddr (Write Address) and RAddr (Read Address) alternate positions in horizontally aligned RAM blocks. For the left-

most RAM blocks, RAddr is on the left and WAddr is on the right. For the right-most RAM blocks, WAddr is on the left and RAddr is tied off. For single-ported RAM, WAddr is the READ/WRITE address port and Din is the (bi-directional) data port. The right-most RAM blocks can be used only for single-ported memories. $\overline{WE}$ and $\overline{OE}$ connect to the vertical express buses in the same column on Plane V_1 and V_2, respectively. WAddr, RAddr, $\overline{WE}$ and $\overline{OE}$ connect to express buses that are full length at array edge.

Reading and writing the 32 x 4 dual-port RAM are independent of each other. Reading the 32 x 4 dual-port RAM is completely asynchronous. Latches are transparent; when Load is logic 1, data flows through; when Load is logic 0, data is latched. Each bit in the 32 x 4 dual-port RAM is also a transparent latch. The front-end latch and the memory latch together and form an edge-triggered flip-flop. When a bit nibble is (Write) addressed and LOAD is logic 1 and $\overline{WE}$ is logic 0, DATA flows through the bit. When a nibble is not (Write) addressed or LOAD is logic 0 or $\overline{WE}$ is logic 1, DATA is latched in the nibble. The two CLOCK muxes are controlled together; they both select CLOCK or they both select “1”. CLOCK is obtained from the clock for the sector-column immediately to the left and immediately above the RAM block. Writing any value to the RAM Clear Byte during configuration clears the RAM, see Figure 5 and Figure 6.

Figure 8. FPGA RAM Connections (One RAM Block)

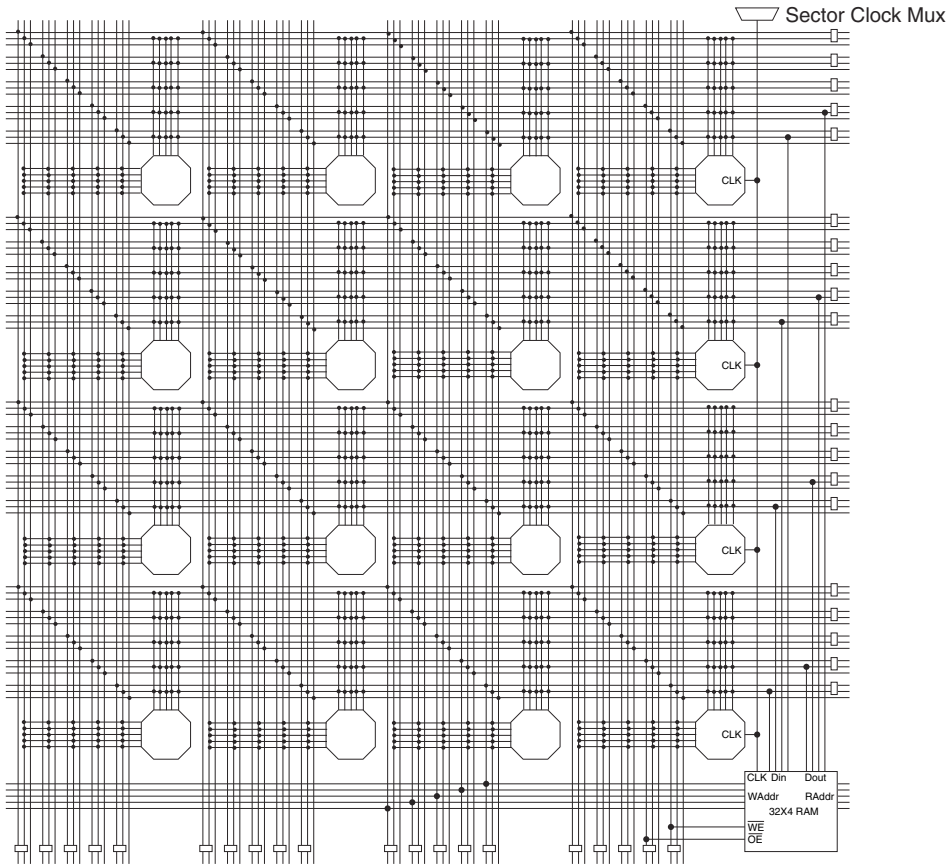
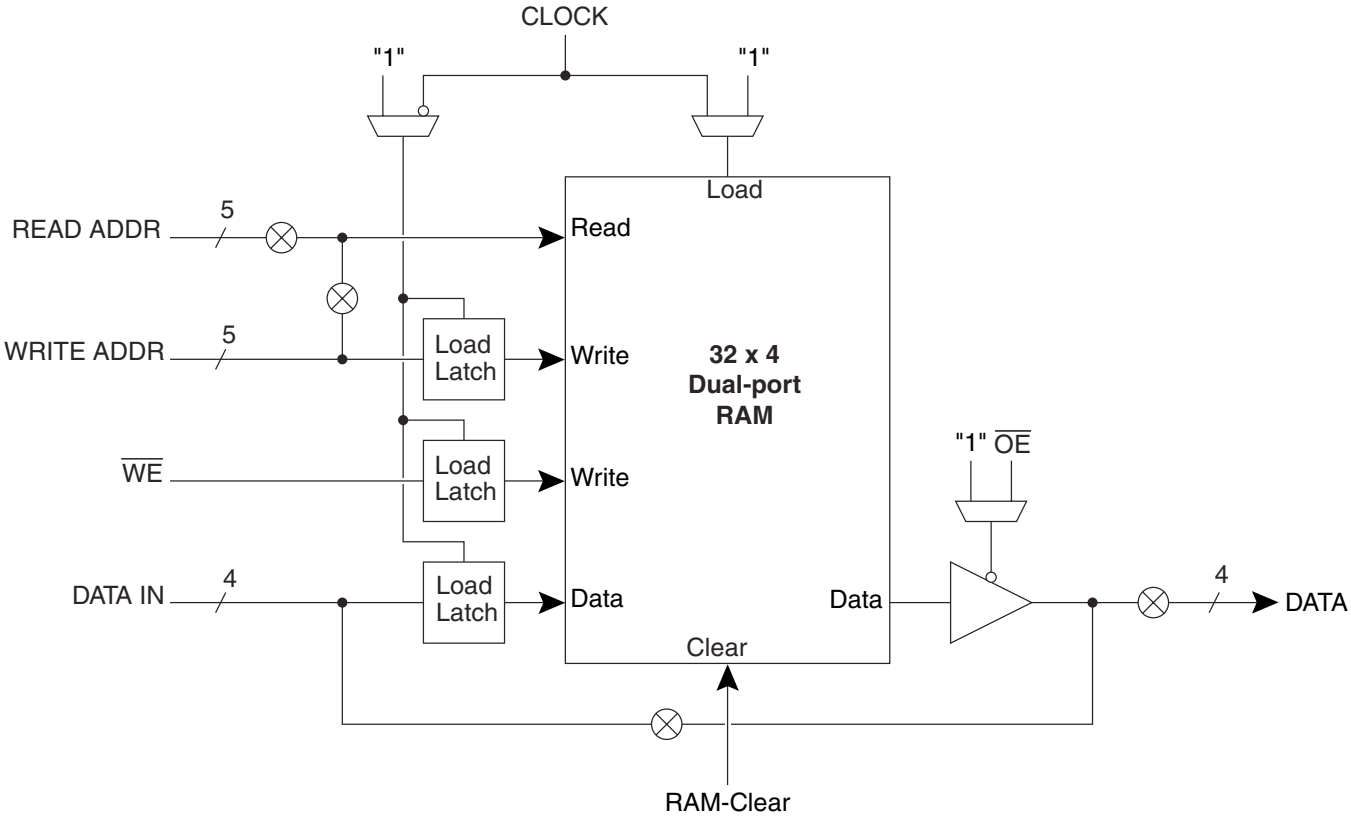
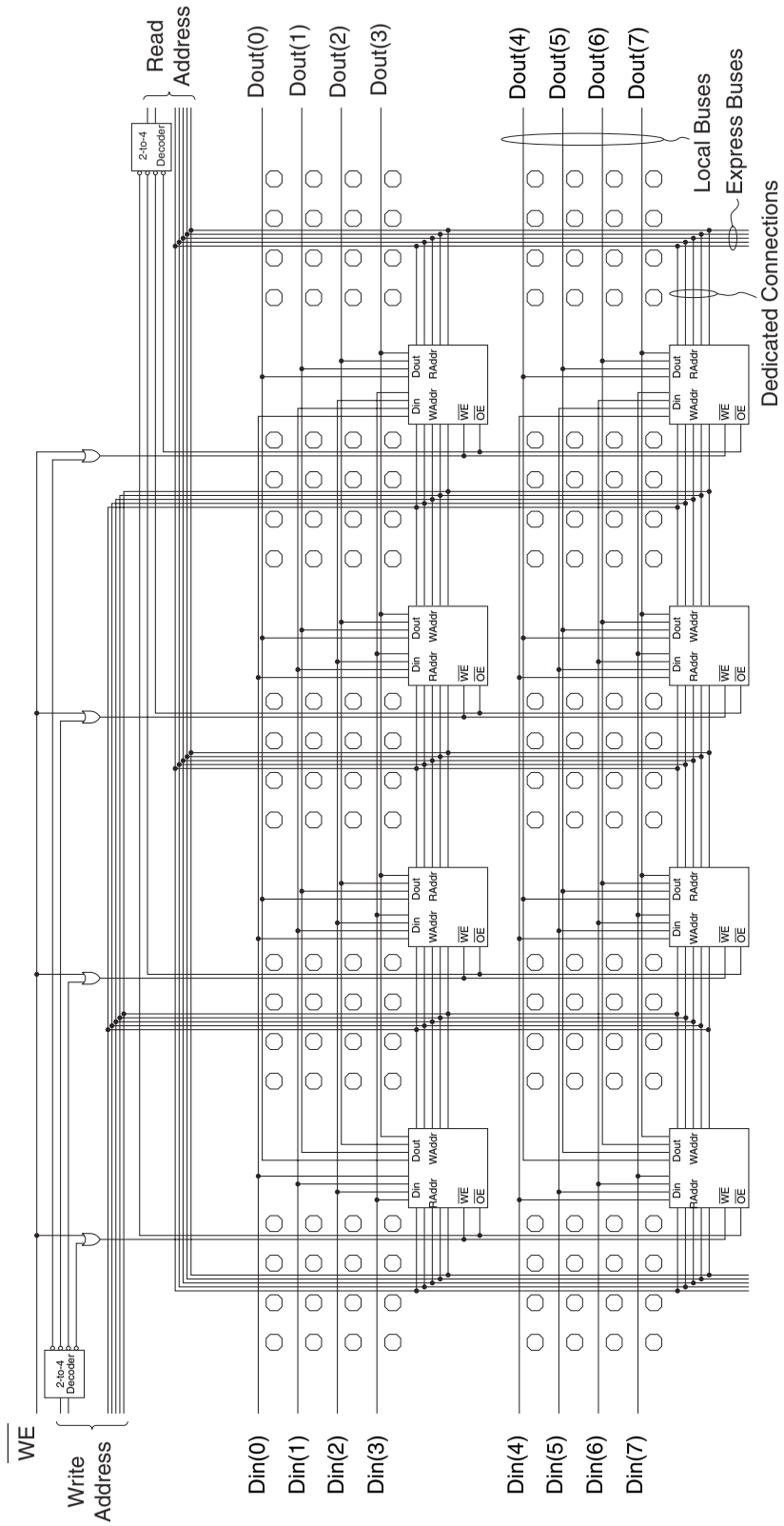


Figure 9. FreeRAM Logic[1]



Note: 1. For dual port, the switches on READ ADDR and DATA OUT would be on. The other two would be off. The reverse is true for single port.

Figure 10. FreeRAM Example: 128 x 8 Dual-ported RAM (Asynchronous)[1]



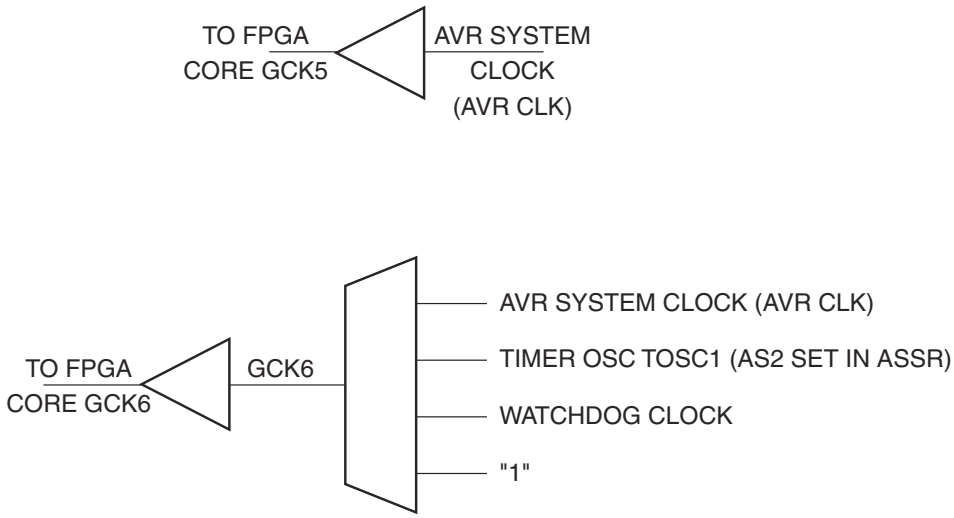
Note: 1. These layouts can be generated automatically using the Macro Generators.

Clocking and Set/Reset

Six of the eight dedicated Global Clock buses (1, 2, 3, 4, 7 and 8) are connected to a dual-use Global Clock pin. In addition, two Global Clock buses (5 and 6) are driven from clock signals generated within the AVR microcontroller core, see Figure 11.

An FPGA core internal signal can be placed on any Global Clock bus by routing that signal to a Global Clock access point in the corners of the embedded core. Each column of the array has a Column Clock selected from one of the eight Global Clock buses. The left edge Column Clock mux has two additional inputs from dual-use pins FCK1, see Figure 8, and FCK2 to provide fast clocking to left-side I/O. Each sector column of four cells can be clocked from a (Plane 4) express bus or from the Column Clock. Clocking to the 4 cells of a sector can be disabled. The Plane 4 express bus used for clocking is half length at the array edge. The clock provided to each sector column of four cells can be either inverted or not inverted. The register in each cell is triggered on a rising clock edge. On power-up, constant “0” is provided to each register’s clock pins. A dedicated Global Set/Reset bus, see Figure 9, can be driven by any USER I/O pad, except those used for clocking, Global or Fast. An internal signal can be placed on the Global Set/Reset bus by routing that signal to the pad programmed as the Global Set/Reset input. Global Set/Reset is distributed to each column of the array. Each sector column of four cells can be Set/Reset by a (Plane 5) express bus or by the Global Set/Reset. The Plane 5 express bus used for Set/Reset is half length at array edge. The Set/Reset provided to each sector column of four cells can be either inverted or not inverted. The function of the Set/Reset input of a register (either Set or Reset) is determined by a configuration bit for each cell. The Set/Reset input of a register is Active Low (logic 0). Setting or resetting of a register is asynchronous. On power-up, a logic 1 (High) is provided by each register, i.e., all registers are set at power-up.

Figure 11. FPGA Clocks from AVR



The FPGA clocks from the AVR are effected differently in the various sleep modes of the AVR, see Table 3.

The source clock into the FPGA GCK5 and GCK6 will determine what happens during the various power-down modes of the AVR.

If the XTAL clock input is used as an FPGA clock (GCK5 or GCK6) in Idle mode, it will still be running. In Power-down/save mode the XTAL clock input will be off.

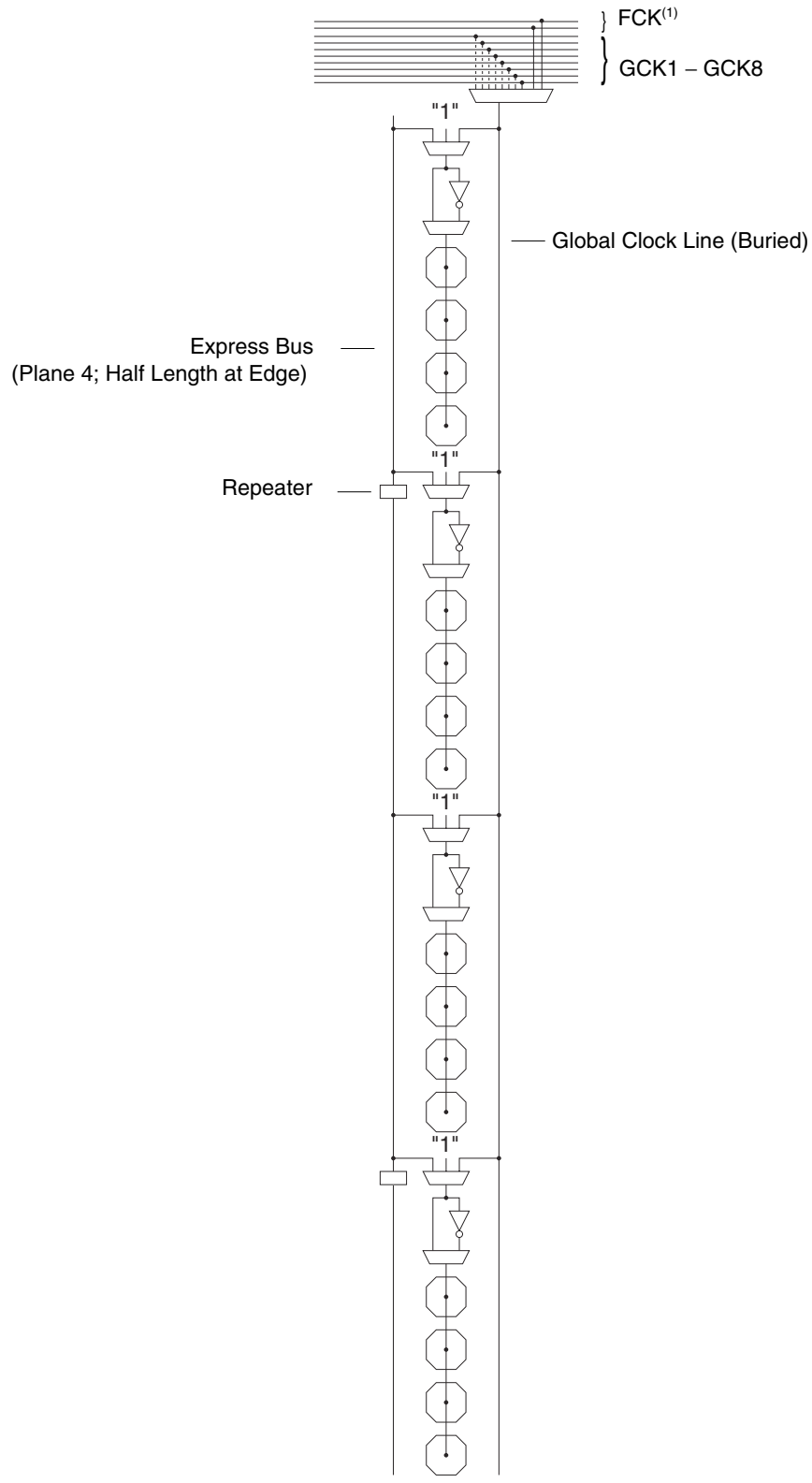
If the TOSC clock input is used as an FPGA clock (GCK6) in Idle mode, it will still be running in Power-save mode but will be off in Power-down mode.

If the Watchdog Timer is used as an FPGA clock (GCK6) and was enabled in the AVR, it will be running in all sleep modes.

Table 3. Clock Activity in Various Modes

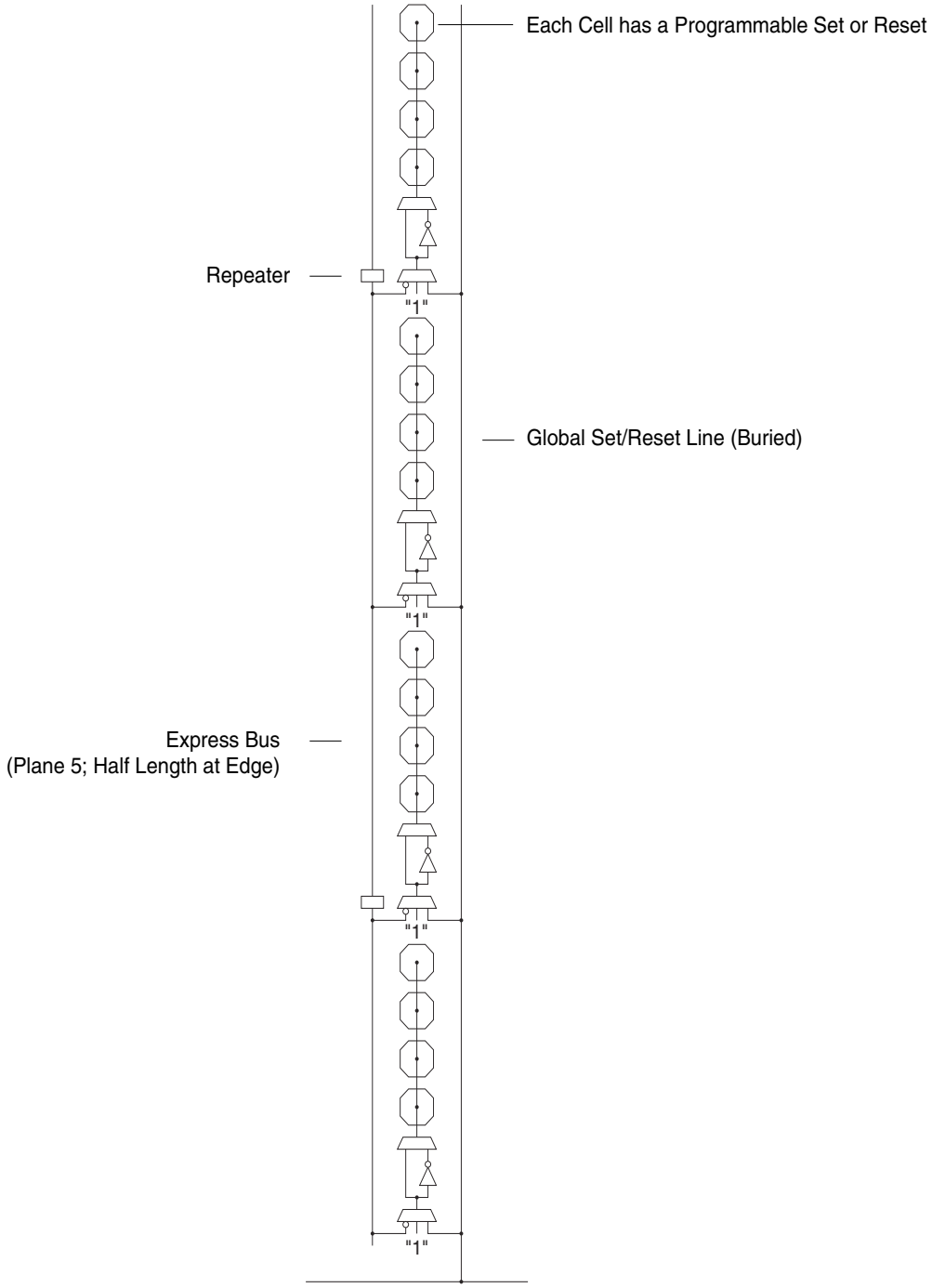
Mode	Clock Source	GCK5	GCK6
Idle	XTAL	Active	Active
	TOSC	Not Available	Active
	WDT	Not Available	Active
Power-save	XTAL	Inactive	Inactive
	TOSC	Not Available	Active
	WDT	Not Available	Active
Power-down	XTAL	Inactive	Inactive
	TOSC	Not Available	Inactive
	WDT	Not Available	Active

Figure 12. Clocking (for One Column of Cells)



Note: 1. Two on left edge column of the embedded FPGA array only.

Figure 13. Set/Reset (for One Column of Cells)



Some of the bus resources on the embedded FPGA core are used as dual-function resources. Table 4 shows which buses are used in a dual-function mode and which bus plane is used. The FPGA software tools are designed to automatically accommodate dual-function buses in an efficient manner.

Table 4. Dual-function Buses

Function	Type	Plane(s)	Direction	Comments
Cell Output Enable	Local	5	Horizontal and Vertical	
FreeRAM Output Enable	Express	2	Vertical	Bus full length at array edge bus in first column to left of RAM block
FreeRAM Write Enable	Express	1	Vertical	Bus full length at array edge bus in first column to left of RAM block
FreeRAM Address	Express	1 - 5	Vertical	Buses full length at array edge buses in second column to left of RAM block
FreeRAM Data In	Local	1	Horizontal	
FreeRAM Data Out	Local	2	Horizontal	
Clocking	Express	4	Vertical	Bus full length at array edge
Set/Reset	Express	5	Vertical	Bus full length at array edge

Figure 14. Primary I/O

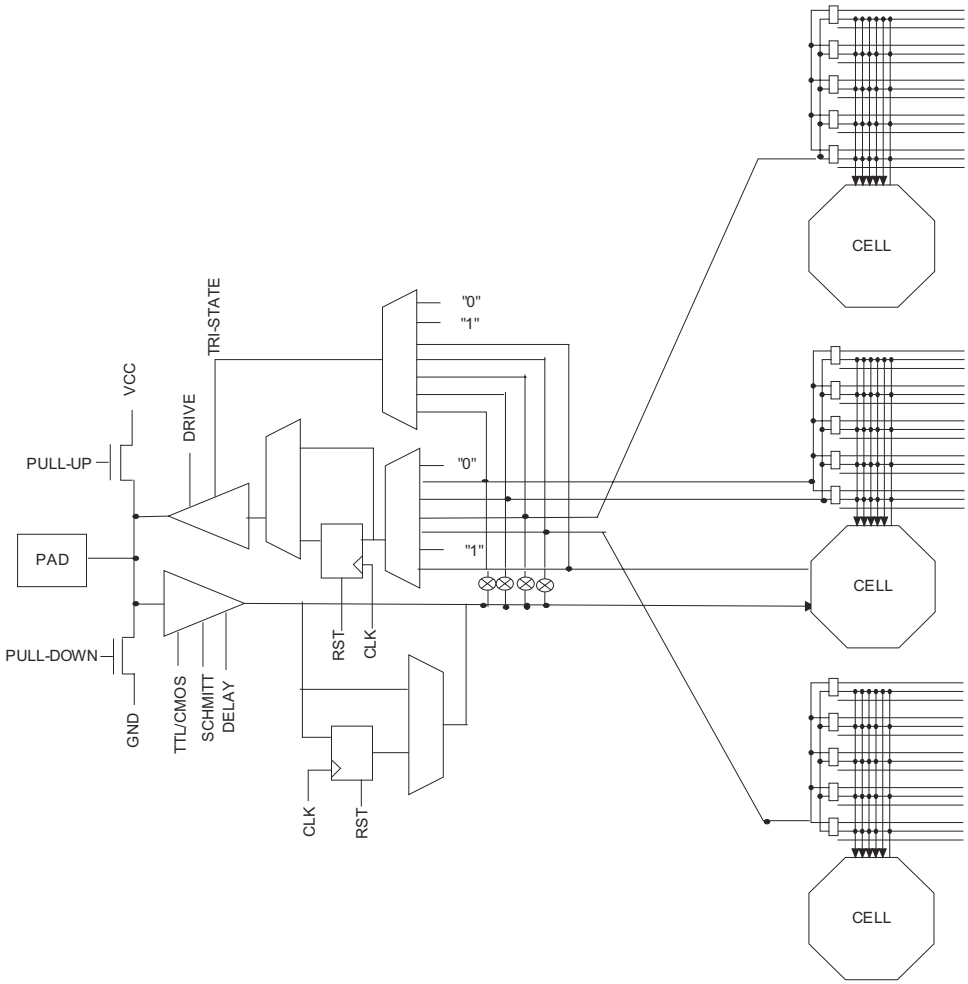


Figure 15. Secondary I/O

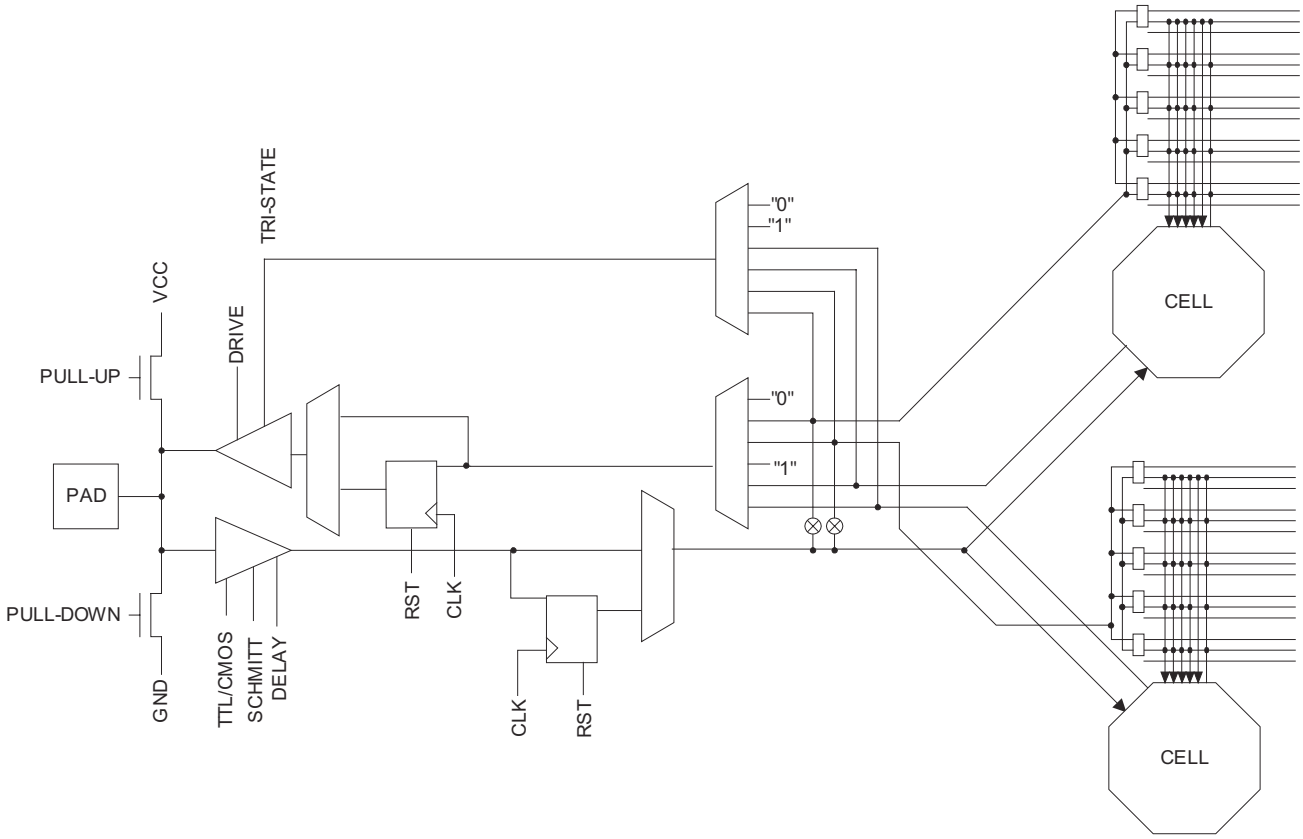


Figure 16. Primary and Secondary I/Os

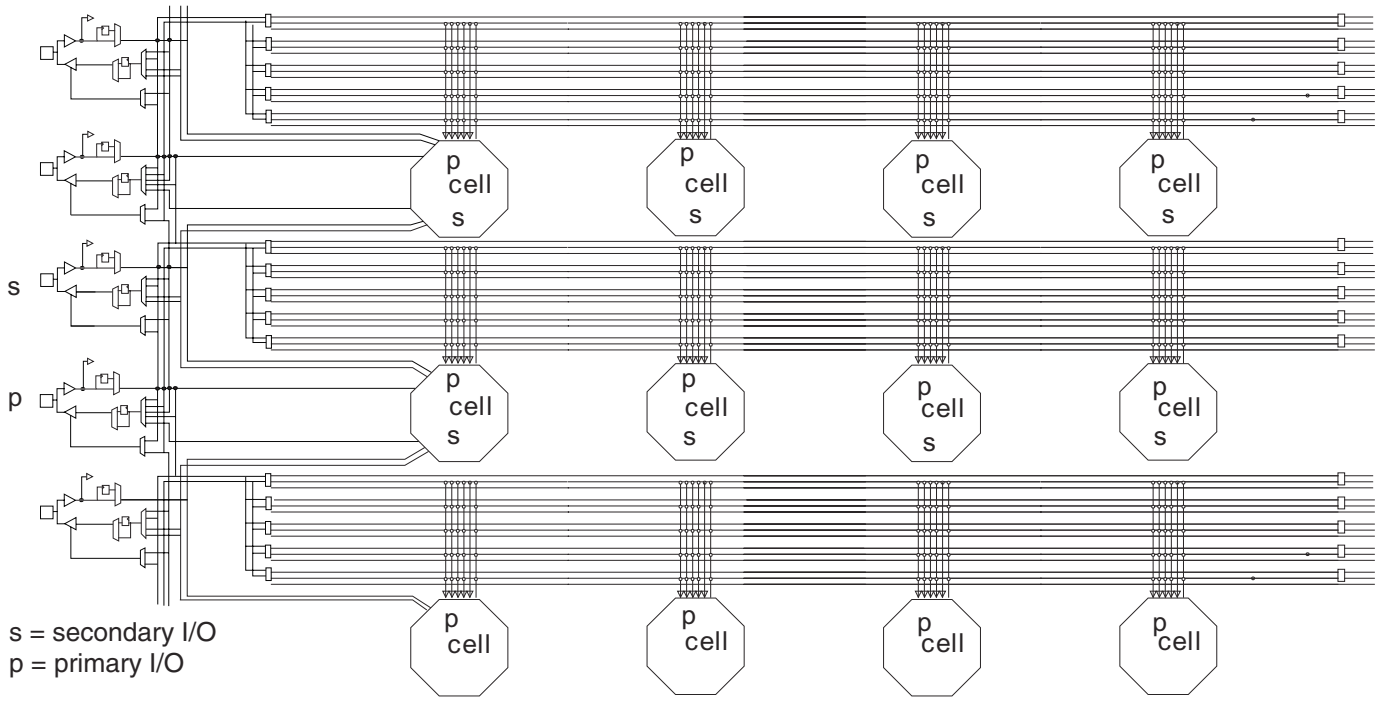
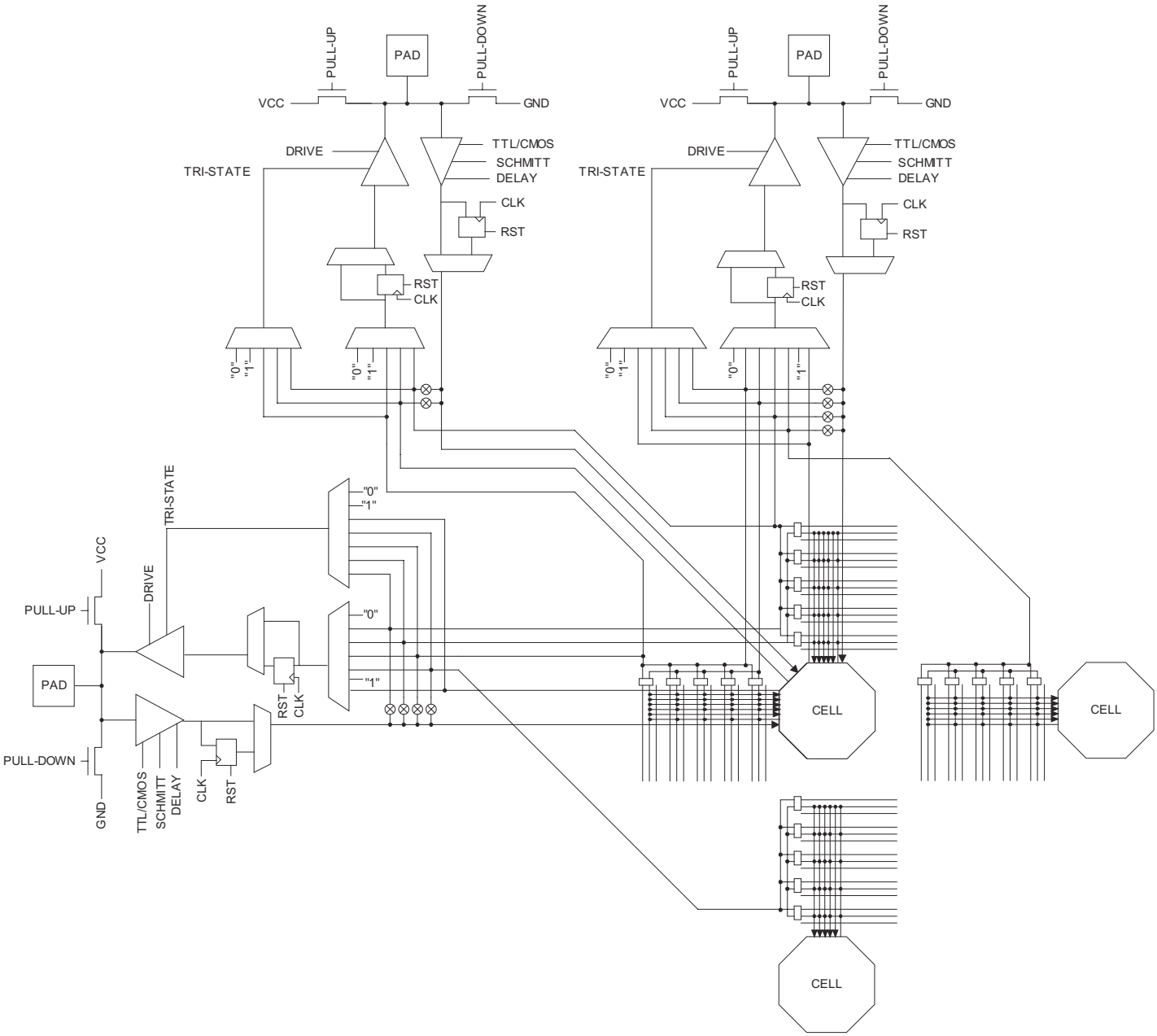


Figure 17. Corner I/Os



FPGA/AVR Interface and System Control

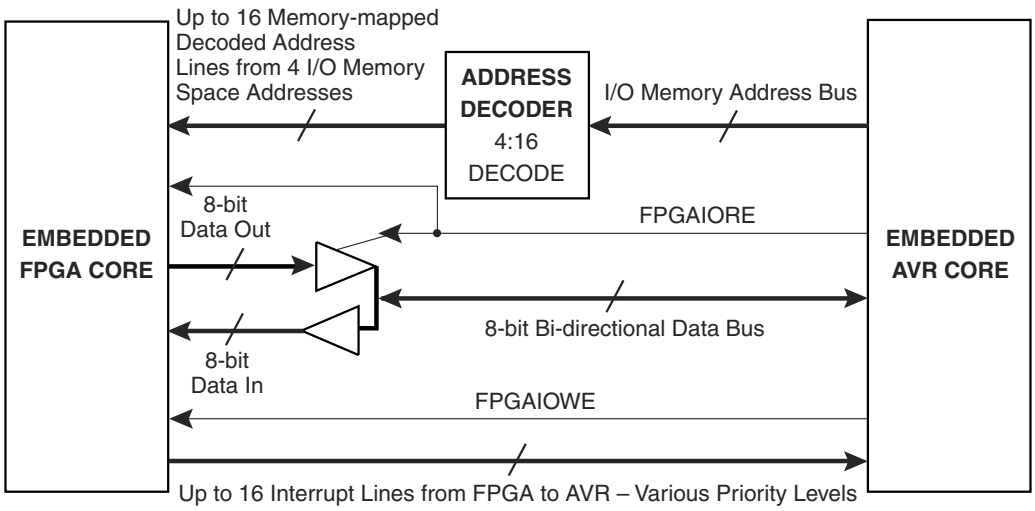
The FPGA and AVR share a flexible interface which allows for many methods of system integration.

- Both FPGA and AVR share access to the 15 ns dual-port SRAM.
- The AVR data bus interfaces directly into the FPGA busing resources, effectively treating the FPGA as a large I/O device. Users have complete flexibility on the types of additional peripherals which are placed and routed inside the FPGA user logic.
- Up to 16 decoded address lines are provided into the FPGA.
- Up to 16 interrupts are available from the FPGA to the AVR.
- The AVR can reprogram the FPGA during operation to create a dynamic reconfigurable system (Cache Logic).

FPGA/AVR Interface– Memory-mapped Peripherals

The FPGA core can be directly accessed by the AVR core, see Figure 18. Four memory locations in the AVR memory map are decoded into 16 select lines (8 for AT94K05) and are presented to the FPGA along with the AVR 8-bit data bus. The FPGA can be used to create additional custom peripherals for the AVR microcontroller through this interface. In addition there are 16 interrupt lines (8 for AT94K05) from the FPGA back into the AVR interrupt controller. Programmable peripherals or regular logic can use these interrupt lines. Full support for programmable peripherals is available within the System Designer tool suite.

Figure 18. FPGA/AVR Interface: Interrupts and Addressing



The FPGA I/O selection is controlled by the AVR. This is described in detail beginning on page 53. The FPGA I/O interrupts are described beginning on page 57.

Program and Data SRAM

Up to 36 Kbytes of 15 ns dual-port SRAM reside between the FPGA and the AVR. This SRAM is used by the AVR for program instruction and general-purpose data storage. The AVR is connected to one side of this SRAM; the FPGA is connected to the other side. The port connected to the FPGA is used to store data without using up bandwidth on the AVR system data bus.

The FPGA core communicates directly with the data SRAM[1] block, viewing all SRAM memory space as 8-bit memory.

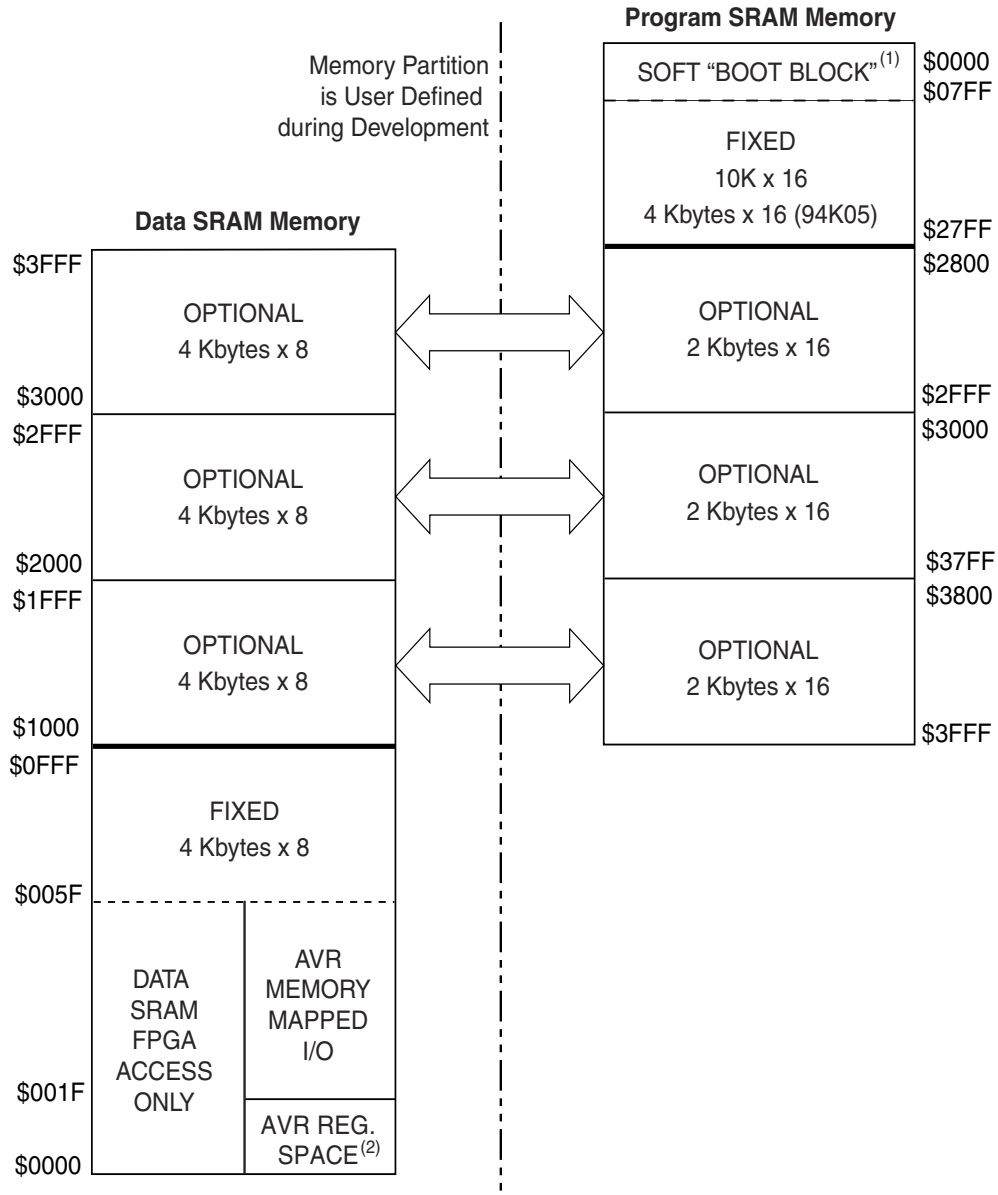
Note: 1. The unused bits for the FPGA-SRAM address must tie to '0' because there is no pull-down circuitry.

For the AT94K10 and AT94K40, the internal program and data SRAM is divided into three blocks: 10 Kbytes x 16 dedicated program SRAM, 4 Kbytes x 8 dedicated data SRAM and 6 Kbytes x 16 or 12 Kbytes x 8 configurable SRAM, which may be swapped between program and data memory spaces in 2 Kbytes x 16 or 4 Kbytes x 8 partitions.

For the AT94K05, the internal program and data SRAM is divided into three blocks: 4 Kbytes 16 dedicated program SRAM, 4 Kbytes x 8 dedicated data SRAM and 6 Kbytes x 16 or 12 Kbytes x 8 configurable SRAM, which may be swapped between program and data memory spaces in 2 Kbytes x 16 or 4 Kbytes x 8 partitions.

The addressing scheme for the configurable SRAM partitions prevents program instructions from overwriting data words and vice versa. Once configured (SCR41:40 – See "System Control Register – FPGA/AVR" on page 30.), the program memory space remains isolated from the data memory space. SCR41:40 controls internal muxes. Write enable signals allow the memory to be safely segmented. Figure 19 shows the FPSLIC configurable allocation SRAM memory.

Figure 19. FPSLIC Configurable Allocation SRAM Memory[1][2]

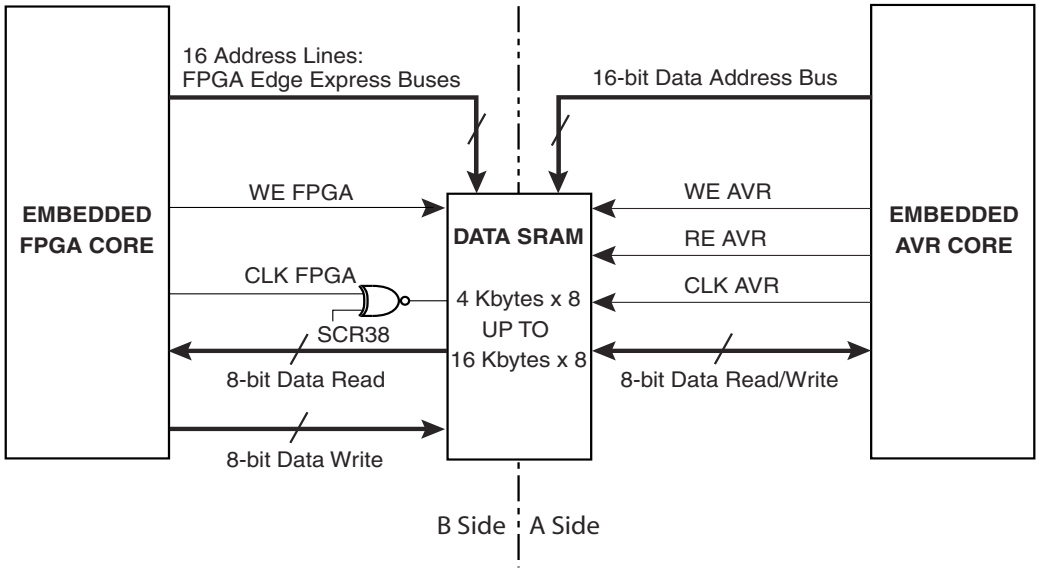


Notes:
1. The Soft "BOOT BLOCK" is an area of memory that is first loaded when the part is powered up and configured. The remainder of the memory can be reprogrammed while the device is in operation for switching functions in and out of memory. The Soft "BOOT BLOCK" can only be programmed by a full device configuration on power-up.
2. The lower portion of the Data memory is not shared between the AVR and FPGA. The AVR uses addresses $0000 - $001F for the AVR CPU general working registers. $001F - $005F are the addresses used for Memory Mapped I/O and store the information in dedicated registers. Therefore, on the FPGA side $0000 - $005F are available for data that is only needed by the FPGA.

Data SRAM Access by FPGA – FPGAFrame Mode

The FPGA user logic has access to the data SRAM directly through the FPGA side of the dual-port memory, see Figure 20. A single bit in the configuration control register (SCR63 – see "System Control Register – FPGA/AVR" on page 30) enables this interface. The interface is disabled during configuration downloads. Express buses on the East edge of the array are used to interface the memory. Full read and write access is available. To allow easy implementation, the interface itself is dedicated in routing resources, and is controlled in the System Designer software suite using the AVR FPGA interface dialog.

Figure 20. Internal SRAM Access – Normal Use



Once the SCR63 bit is set there is no additional read enable from the FPGA side. This means that the read is always enabled. You can also perform a read or write from the AVR at the same time as an FPGA read or write. If there is a possibility of a write address being accessed by both devices at the same time, the designer should add arbitration to the FPGA Logic to control who has priority. In most cases the AVR would be used to restrict access by the FPGA using the FMXOR bit, see "Software Control Register – SFTCR" on page 51. You can read from the same location from both sides simultaneously.

SCR bit 38 controls the polarity of the clock to the SRAM from the AT40K FPGA.

SRAM Access by FPGA/AVR

This option is used to allow for code (Program Memory) changes.

Accessing and Modifying the Program Memory from the AVR

The FPSLIC SRAM is up to 36 x 8 Kbytes of dual port, see Figure 19):

- The A side (port) is accessed by the AVR.
- The B side (port) is accessed by the FPGA/Configuration Logic.
- The B side (port) can be accessed by the AVR with ST and LD instructions in DBG mode for code self-modify.

Structurally, the [(n • 2) Kbytes 8] memory is built from (n)2 Kbytes 8 blocks, numbered SRAM0 through SRAM(n).

A Side

The A side is partitioned into Program memory and Data memory:

- Program memory is 16-bit words.
- Program memory address $0000 always starts in the highest two SRAMs (n - 1, n) [SRAMn - 1 (low byte) and SRAMn (high byte)] (SRAM labels are for layout, the addressing scheme is transparent to the AVR PC).
- System configuration determines the higher addresses for program memory:
  - SCR bits 41 = 0 : 40 = 0, program memory extended from $2800 - $3FFF
  - SCR bits 41 = 0 : 40 = 1, program memory extended from $2800 - $37FF
  - SCR bits 41 = 1 : 40 = 0, program memory extended from $2800 - $2FFF
  - SCR bits 41 = 1 : 40 = 1, no extra program memory
- Extended program memory is always lost to extended data memory from SRAM2/3 down to SRAM6/7, see Table 5.

Table 5. AVR Program Decode for SRAM 2:7 (16K16)

Address Range	SRAM	Comments
$3FFF - $3800 $3FFF - $3800	02 03	CR41:40 = 00
$37FF - $3000 $37FF - $3000	04 05	CR41:40 = 00,01
$2FFF - $2800 $2FFF - $2800	06 07	CR41:40 = 00,01,10
$27FF - $2000 $27FF - $2000 $1FFF - $1800 $1FFF - $1800 $17FF - $1000 $17FF - $1000 $0FFF - $0800 $0FFF - $0800 $07FF - $0000 $07FF - $0000	08 09 10 11 12 13 14 15 16 n = 17	AVR Program Read-only AVR Program Read-only AVR Program Read-only AVR Program Read-only AVR Program Read-only AVR Program Read-only AVR Program Read-only AVR Program Read-only AVR Program Read-only AVR Program Read-only

- Data memory is 8-bit words.
- Data memory address $0000 always starts in SRAM0 (SRAM labels are for layout, the addressing scheme is transparent to AVR data read/write).
- System configuration determines the higher address for data memory:
  - SCR bits 41 = 0 : 40 = 0, no extra data memory
  - SCR bits 41 = 0 : 40 = 1, data memory extended from $1000 - $1FFF
  - SCR bits 41 = 1 : 40 = 0, data memory extended from $1000 - $2FFF
  - SCR bits 41 = 1 : 40 = 1, data memory extended from $1000 - $3FFF
- Extended data memory is always lost to extended program memory from SRAM7 up to SRAM2 in 2 x SRAM blocks, see Table 6.