AT40K IP Generator Guide

June 2002

Table of Contents

IP Core Generator

Features

- **Schematic Generation (AT40K & AT40KAL)**
- **Symbol Generation (AT40K & AT40KAL)**
- **Hard Macro Generation**
- **User-defined Macro Name**
- **User-defined Pins**
- **User-defined Libraries**
- **Flat Netlist Generation for Simulation**
- **Check-Out Macro for Modification**

Introduction

The IP Core Generator is designed to utilize many innovative features of Atmel's AT40K, AT40KAL and AT94K Field Programmable System Level Integrated Circuit (FPSLIC™) architecture. High-speed custom logic functions can be rapidly created, resulting in significantly improved performance for computation-intensive applications. Since the AT40K architecture is the foundation for AT40KAL and AT94K Series devices, the cores generated are compatible across all three families.

The core generator facilitates quick and easy implementation of over 50 logic and memory cores such as multipliers, adders, accumulators, dual-port RAM, FIFO etc. The design tool will automatically create a hard layout with area information and worst-case speed, it will also generate the schematic, the symbol for schematic-based designs, and a back-annotated VHDL or Verilog netlist for simulation.

The IP cores are parameterized, which means that user-defined macros can be generated by specifying the required parameters. The tool allows designers to create their own user libraries and store the cores generated in the library for current and future designs.

Programmable SLI
AT40K
AT40KAL
AT94K

Application Note

Rev. 2421C–FPSLI–02/02

Statistical Summary

Table 1 is a quick reference summary of all of the IP Core generators. Delay values are given for 1 speed grade on the AT94K/AT40KAL architecture.

Table 1. Statistical Summary

Generator	Name	Speed (Mhz)	Delay (ns)	Cells	Size (x * y)	Latency (Clocks)
Absolute Value	abs16	54.8	18.2	15	1 x 15	0
	abs8	115.7	8.6	7	1 x 7	0
Accumulator	acc16	38.7	25.8	36	2 x 18	0
	acc8	61.6	16.2	20	2 x 10	0
Adder – Carry Select	acs16	47	21.3	48	3 x 19	0
Adder – Ripple Carry	arc16	49.9	20.1	16	1 x 16	0
	arc8	95.6	10.5	8	1 x 8	0
Comparator	com16	56.1	17.8	16	8 x 2	0
	com8	128	7.8	8	4 x 2	0
Counter – Johnson	cjo16	98	10.2	16	1 x 16	0
	cjo8	120.2	8.3	8	1 x 8	0
Counter – LFSR	lfs16	94.7	10.6	16	1 x 16	0
	lfs8	110.6	9	8	1 x 8	0
Counter – PreScaled	cps16	45.2	22.1	17	1 x 17	0
	cps8	82.1	12.2	9	1 x 9	0
Counter – Ripple Carry	crc16	45.7	21.9	16	1 x 16	0
	crc8	83.6	12.6	8	1 x 8	0
Counter – Terminal	ctr16	85	11.8	6	1 x 16	0
	ctr8	95	10.5	5	1 x 5	0
Decoder	dec16	492.6	2	16	1 x 16	0
	dec8	492.6	2	4	1 x 4	0
Deductor	ded16	38.7	25.9	34	2 x 18	0
	ded8	61.5	16.3	18	2 x 10	0
Flip Flop – D Type	fdt16	598.8	1.7	16	1 x 16	0
	fdt8	598.8	1.7	8	1 x 8	0
Flip Flop – Toggle	fft16	598.8	1.7	16	1 x 16	0
	fft8	598.8	1.7	8	1 x 8	0
FIFO	fif16	46	21.7	19	7 x 6	0
	fif8	45.8	21.8	16	7 x 5	0
Gray Code	gra16	543.5	1.8	8	1 x 8	0
	gra8	543.5	1.8	4	1 x 4	0

Table 1. Statistical Summary

Generator	Name	Speed (Mhz)	Delay (ns)	Cells	Size (x * y)	Latency (Clocks)
Incrementor/ Decrementor by 1	inc16	39.4	25.4	17	1 x 17	0
	inc8	64.8	15.4	9	1 x 9	0
Incrementor/ Decrementor by Value	inv16	37.6	26.6	17	1 x 17	0
	inv8	60	16.7	9	1 x 9	0
Transparent Latch	ldt16	719.4	1.4	16	1 x 16	0
	ldt8	719.4	1.4	8	1 x 8	0
Logic Gates	log16	138.9	7.2	6	4 x 2	0
	log8	304.9	3.3	3	3 x 2	0
Look-Up Table	lut16	492.6	2	16	1 x 16	0
	lut8	492.6	2	8	1 x 8	0
Multiplier – Serial Parallel	msp16	108.3	9.2	32	2 X 16	0
	msp8	108.3	9.2	16	2 X 8	0
Multiplier – Signed	mls16	22.2	45.1	189	17 X 17	0
	mls8	40.8	24.5	81	9 X 9	0
Multiplier – Signed, 1 Stage of Pipelining	mps16	33.8	29.6	306	17 x 18	1
	mps8	68.6	14.6	90	7 X 10	1
Multiplier – Unsigned	mlu16	22.8	43.8	256	16 x 16	
	mlu8	42.9	23.3	64	8 x 8	
Multiplier – Unsigned, 1 Stage of Pipelining	mup16	38.7	25.8	272	16 x 7	1
	mup8	84.9	11.8	72	8 x 9	
Mux	mux_16	140.6	7.1	16	8 x 2	
	mux_8	154.1	6.5	8	4 x 2	0
Negate Function	nef16	51.4	19.4	16	1 x 16	0
	nef8	101.6	9.8	8	1 x 8	0
Pulse Generator – Fixed	pls16	141.5	7.1	5	1 x 5	0
	pls8	171.5	5.6	4	1 x 4	0
Pulse Generator – Loadable	pll16	25.2	39.8	48	3 x 17	0
	pll8	47.9	20.9	24	3 x 9	0
RAM – Dual Port	rdp16	283.3	3.5	1	7 x 2	0
	rdp8	283.3	3.5	1	7 x 2	0
RAM – Single Port	rsp16	283.3	3.5	1	7 x 2	0
	rsp8	283.3	3.5	1	7 x 2	0

Table 1. Statistical Summary

Generator	Name	Speed (Mhz)	Delay (ns)	Cells	Size (x * y)	Latency (Clocks)
ROM	rom16	492.6	2	8	8 x 1	0
	rom8	492.6	2	8	4 x 2	0
Subtractor – Carry Select	scs16	47	21.3	48	3 x 19	0
Subtractor – Ripple Carry	src16	49.9	20.1	16	1 x 16	0
	src8	95.6	10.5	8	1 x 8	0
Shift Register	sre16	598.8	1.7	16	1 x 16	0
	sre8	598.8	1.7	8	1 x 8	0

PSLI Macro Library

Features

- **Functional Macros**
- **Dynamic Macros**

Description

The Programmable System Level Integrated (PSLI) library of components can be divided into 2 types of macros: functional and dynamic. Functional macros are components with fixed functionality, such as the 2-input AND gate. Dynamic macros are designed to allow user specification of any desired functionality attached as an attribute, via an equation string, on the symbol. This should be used only when a specific function for a core cell is required. Designs targeted to PSLI devices can use a mix of dynamic and functional macros.

The use of the PSLI Macros depends on the design entry method being used. For schematic entry, all of the components shown in this library could be used. For HDL entry (VHDL® or Verilog®), these components would be used by the Synthesis tool when synthesizing from VHDL or Verilog to the PSLI device.

Programmable SLI

AT40K

AT40KAL

AT94K

Application Note

Rev. 2448A–12/01

Functional Macros

Logical Function	Description	Area (x * y)
Gates – NAND		
ND2	2-input NAND	1 x 1
ND2I1	2-input NAND with 1 input inverted	1 x 1
ND2I2	2-input NAND with 2 inputs inverted	1 x 1
ND3	3-input NAND	1 x 1
ND3I1	3-input NAND with 1 input inverted	1 x 1
ND3I2	3-input NAND with 2inputs inverted	1 x 1
ND3I3	3-input NAND with 3 inputs inverted	1 x 1
ND4	4-input NAND	1 x 1
ND4I1	4-input NAND with 1 input inverted	1 x 1
ND4I2	4-input NAND with 2 inputs inverted	1 x 1
ND4I3	4-input NAND with 3 inputs inverted	1 x 1
ND4I4	4-input NAND with 4 inputs inverted	1 x 1
Gates – AND		
AN2	2-input AND	1 x 1
AN2I1	2-input AND with 1 input inverted	1 x 1
AN2I2	2-input AND with 2 inputs inverted	1 x 1
AN3	3-input AND	1 x 1
AN3I1	3-input AND with 1 input inverted	1 x 1
AN3I2	3-input AND with 2 inputs inverted	1 x 1
AN3I3	3-input AND with 3 inputs inverted	1 x 1
AN4	4-input AND	1 x 1
AN4I1	4-input AND with 1 input inverted	1 x 1
AN4I2	4-input AND with 2 inputs inverted	1 x 1
AN4I3	4-input AND with 3 inputs inverted	1 x 1
AN4I4	4-input AND with 4 inputs inverted	1 x 1
Gates – OR		
OR2	2-input OR	1 x 1
OR2I1	2-input OR with 1 input inverted	1 x 1
OR2I2	2-input OR with 2 inputs inverted	1 x 1
OR3	3-input OR	1 x 1
OR3I1	3-input OR with 1 input inverted	1 x 1
OR3I2	3-input OR with 2 inputs inverted	1 x 1
OR3I3	3-input OR with 3 inputs inverted	1 x 1

Functional Macros (Continued)

Logical Function	Description	Area (x * y)
OR4	4-input OR	1 x 1
OR4I1	4-input OR with 1 input inverted	1 x 1
OR4I2	4-input OR with 2 inputs inverted	1 x 1
OR4I3	4-input OR with 3 inputs inverted	1 x 1
OR4I4	4-input OR with 4 inputs inverted	1 x 1
Gates – NOR		
NR2	2-input OR	1 x 1
NR2I1	2-input OR with 1 input inverted	1 x 1
NR2I2	2-input OR with 2 inputs inverted	1 x 1
NR3	3-input OR	1 x 1
NR3I1	3-input OR with 1 input inverted	1 x 1
NR3I2	3-input OR with 2 inputs inverted	1 x 1
NR3I3	3-input OR with 3 inputs inverted	1 x 1
NR4	4-input OR	1 x 1
NR4I1	4-input OR with 1 input inverted	1 x 1
NR4I2	4-input OR with 2 inputs inverted	1 x 1
NR4I3	4-input OR with 3 inputs inverted	1 x 1
NR4I4	4-input OR with 4 inputs inverted	1 x 1
Gates – XOR		
XO2	2-input XOR	1 x 1
XO3	3-input XOR	1 x 1
XO4	4-input XOR	1 x 1
XN2	2-input XNOR	1 x 1
XN3	3-input XNOR	1 x 1
XN4	4-input XNOR	1 x 1
Inverters		
INV	Inverter	1 x 1
Constants		
ONE	Logic one	1 x 1
ZERO	Logic zero	1 x 1
Multiplexers		
MUX2	2 to 1 multiplexer	1 x 1
MUX3	3 to 1 multiplexer	1 x 2
Latches		
LD	D latch transparent High	1 x 1

Functional Macros (Continued)

Logical Function	Description	Area (x * y)
LDRA	D latch transparent High, reset Low	1 x 1
LDSA	D latch transparent High, set Low	1 x 1
LDE	D latch transparent High with enable	1 x 1
Flip-Flops		
FD	D flip-flop	1 x 1
FDRA	D flip-flop, asynchronous reset Low	1 x 1
FDSA	D flip-flop, asynchronous set Low	1 x 1
FDE	D flip-flop with enable	1 x 1
FDRAE	D flip-flop with enable, asynchronous reset Low	1 x 1
FDSAE	D flip-flop with enable, asynchronous set Low	1 x 1
FJK	JK flip-flop	1 x 1
FJKRA	JK flip-flop, asynchronous reset Low	1 x 1
FJKSA	JK flip-flop, asynchronous set Low	1 x 1
Arithmetic Functions		
FA	1-bit full adder	1 x 1
MULT	1-bit multiplier	1 x 1
Tri-State functions		
BUFZ	Tri-state buffer	1 x 1
HZ	Bus driver High or Z	1 x 1
LZ	Bus driver Low or Z	1 x 1
RAM Macros		
RAMS	32 x 4 asynchronous single-port RAM	1 x 1
RAMD	32 x 4 asynchronous dual-port RAM	1 x 1
RAMDSYNC	32 x 4 synchronous dual-port RAM	1 x 1
RAMSSYNC	32 x 4 synchronous single-port RAM	1 x 1

Dynamic Macros

This section describes the dynamic macros for the PSLI. The macros are provided to give the user better control over the implementation of specific functions in a single core cell. It can also be used to simplify the design entry process. Pre-defined attributes with user designated values are used to exploit the logic capabilities of the PSLI core cell. The different pre-defined attributes that can be specified for the dynamic macros(1) are listed below:

- FUNCTIONG
- FUNCTIONH
- CLOCKEDGE
- RSFUNCTION
- RSPOLARITY
- PRESERVE

Note: 1. When dynamic macros are used in a design, the circuit cannot be simulated directly. The circuit must be run through to *Initial Placement* before a functional netlist can be generated.

FUNCTIONG

Description Specifies the equation to be implemented as the G output of the macro.

Value An equation string of one to four variables. Details on the equation syntax are explained in “Equation Syntax” on page 10 of this document.

FUNCTIONH

Description Specifies the equation to be implemented as the H output of the macro.

Value An equation string of one to four variables. Details on the equation syntax are explained in “Equation Syntax” on page 10 of this document.

CLOCKEDGE

Description Specifies the rising edge or falling edge trigger on the clock to which the register responds, see Table 1.

Table 1. CLOCKEDGE Function

Value	Explanation
RISING	Positive edge trigger on the register CLK pin
FALLING	Negative edge trigger on the register CLK pin

RSFUNCTION

Description The register in the PSLI core cell can provide set or reset functions through the RS pin on the dynamic macros, see Table 2.

Table 2. RSFUNCTION

Value	Explanation
RESET	RS functions as reset pin
SET	RS functions as set pin

RSPOLARITY

Description

Specifies the polarity of the RS pin, see Table 3.

Table 3. RSPOLARITY Function

Value	Explanation
HIGH	Active High RS pin
LOW	Active Low RS pin

PRESERVE

Description

Specifies if the component should be preserved by the mapper during technology mapping, see Table 4.

Table 4. PRESERVE Function

Value	Explanation
YES	Do not touch the macro during mapping
NO	Macro can be flattened during mapping

Equation Syntax

The equation string attached to the FUNCTIONG and FUNCTIONH attributes describes the combinatorial behavior of the respective outputs of the core cell. To register and/or tri-state the output of the equation, the correct registered or tri-stated dynamic macros must be used from the library. The equation string is a multi-level sum-of-products equation built using the operators shown in Table 5.

Table 5. Equation Operators

Operators	Description
^, ~	Logical NOT
*, &	Logical AND
|, +	Logical OR
#, @	Logical XOR

The dynamic macro input port names, called A, B, C, and D, are the variables used in the equation. Unconnected input ports are allowed on dynamic macro instances. However, an error will be generated if a port name used in the equation string is not connected to a net. The CLK, RS and OE pins on the register and tri-state dynamic macros should not be used in the equation string.

Table 6 describes the dynamic macros available in the PSLI library.

Table 6. Logical Function

Logical Function	Description
FGEN1	*n* input function generator ($1 \le n \le 4$)
FGEN1F	*n* input function generator with combinatorial feedback ($1 \le n \le 3$)
FGEN1FT	*n* input function generator with combinatorial feedback followed by tri-state buffer ($1 \le n \le 3$)
FGEN1R	*n* input function generator followed by a register ($1 \le n \le 4$)
FGEN1RF	*n* input function generator with registered feedback ($1 \le n \le 3$)
FGEN1RFT	*n* input function generator with registered feedback followed by tri-state buffer($1 \le n \le 3$)
FGEN1RT	*n* input function generator followed by a register and tri-state buffer ($1 \le n \le 4$)
FGEN1T	*n* input function generator followed by a tri-state buffer ($1 \le n \le 4$)
FGEN2	Two *n* input function generators ($1 \le n \le 3$)
FGEN2F	Two *n* input function generators with combinatorial feedback on 1-output($1 \le n \le 2$)
FGEN2FT	Two *n* input function generators with combinatorial feedback followed by tri-state buffer on 1-output ($1 \le n \le 2$)
FGEN2R	Two *n* input function generators with 1-output registered and the other combinatorial ($1 \le n \le 3$)
FGEN2RF	Two *n* input function generators with 1-output registered and feedback ($1 \le n \le 2$)
FGEN2RFT	Two *n* input function generators with 1-output registered, tri-stated and feedback ($1 \le n \le 2$)
FGEN2RT	Two *n* input function generators with 1-output registered and tri-stated ($1 \le n \le 3$)
FGEN2T	Two *n* input function generator with 1-output tri-stated ($1 \le n \le 3$)
MGEN[1]	Two *3*-input function generators
MGENR	Two 3-input function generators with 1-output registered
MGENRT	Two 3-input function generators with 1-output registered and tri-stated
MGENT	Two 3-input function generator with 1-output tri-stated

Note: 1. The MGEN macros are special case macros (typically used in multipliers) with an upstream AND gate feeding the Look-Up Tables.

I/O Macros

This section covers the I/O macros available for the PSLI. In order to specify additional functionality on the I/O, various properties or attributes are used. Different pre-defined attributes can be set on the I/O pads in the PSLI library. Attributes are used on the I/O macros to select the threshold levels, different slew rates etc. The pre-defined attributes that can be specified on the I/O macros are listed below:

- THRESHOLD
- SCHMITT
- SLEWRATE
- EXTRADELAY

THRESHOLD

Description

Specifies the threshold level on the input buffers, see Table 7.

Table 7. THRESHOLD Macro

Value	Explanation
CMOS	CMOS on input
TTL	TTL on input

SCHMITT

Description

Specifies whether a Schmitt trigger circuit on the input pads should be enabled or disabled. The Schmitt trigger is a regenerative comparator circuit to improve the rise and fall times (leading and trailing edges) of the incoming signal, see Table 8.

Table 8. SCHMITT Trigger

Value	Explanation
ENABLE	Enable the schmitt trigger circuit
DISABLE	Disable the schmitt trigger circuit

SLEWRATE

Description

Specifies the output drive, see Table 9.

Table 9. SLEWRATE Macro

Value	Explanation
FAST	Full drive (20 mA buffer)
MEDIUM	Medium drive (14 mA buffer)
SLOW	Standard drive (6 mA buffer)

EXTRADELAY

Description

The input buffers in the PSLI library can have four different intrinsic delays. This attribute lets the user specify an extra delay on the input signal to meet any data hold requirements. A value of ‘0’ provides no extra delay above the intrinsic delay of the input buffer and a value of ‘1’ allows an extra delay of approximately 1 ns above the intrinsic delay, see Table 10 and Table 11.

Table 10. Intrinsic Delays

Value	Explanation
0	No extra intrinsic delay
1	Extra intrinsic delay of approximately 1 ns
3	Extra intrinsic delay of approximately 3 ns
5	Extra intrinsic delay of approximately 5 ns

Table 11. Logical Functions

Logical Function	Description
IBUF	Input buffer
OBUF	Output buffer
OBUFE	Output buffer with active High enable
OBUFOD	Output buffer open drain
OBUFOS	Output buffer open source
BIBUF	Bi-directional buffer
BIBUFOD	Bi-directional buffer open drain
BIBUFOS	Bi-directional buffer open source
GCLKBUF	Global clock buffer
RSBUF	Global reset buffer
FCLKBUF	Fast clock buffer

IP Core Generator: Absolute Value

Features

- **Accessible from the Macro Generator Dialog and HDLPlanner™ – Included in IDS for FPGA Devices and System Designer™ for AT94K FPSLIC™ Devices**
- **Optional Overflow Pin**
- **Variable Width for Input and Output Vectors**

Description

The function of the Absolute Value generator can be explained as follows:

if DATA = $-2^{(Width - 1)}$, then

    OVERFLOW = 1, RESULT = 0

else if DATA < 0, then

    RESULT = - DATA

else

    RESULT = DATA

DATA must always represent a two's complement and the RESULT is always a positive number.

Parameters

Parameter	Value	Explanation
Overflow	Boolean	Create with overflow pin
Width	Integer > 1	Width of input and output vectors

Pins

Type	Name	Option	Explanation
In	DATA[Width - 1:0]	No	Input data
Out	RESULT[Width - 1:0]	No	Absolute value of data (number of outputs will be width - 1 if overflow is used)
Out	OVERFLOW	Yes	True if Data = $-2^{(Width - 1)}$

Programmable SLI
AT40K
AT40KAL
AT94K

Application Note

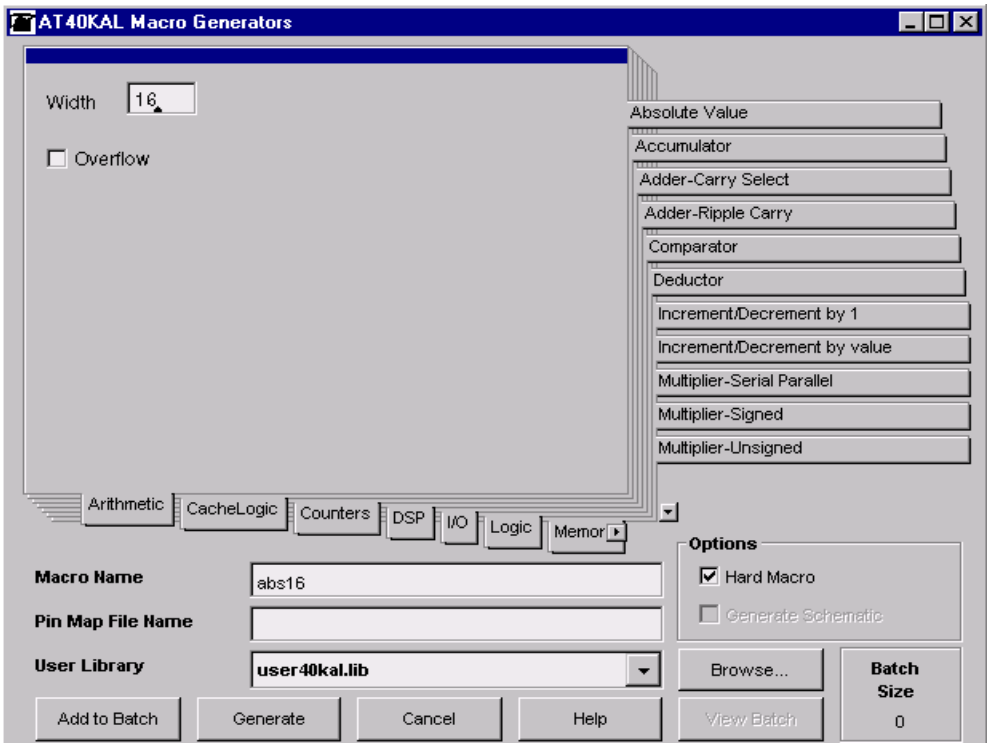
Rev. 2423B–FPSLI–01/16/02

Statistics

Device	Name	Speed (MHz)	Delay (ns)	Cells	Size (x * y)
AT40K	abs16	38.88	25.8	15	1 x 15
AT40K	abs8	79.4	12.6	7	1 x 7
AT94K/ AT40KAL	abs16	54.8	18.2	15	1 x 15
AT94K/ AT40KAL	abs8	115.7	8.6	7	1 x 7

Figure 1 shows an example of the abs16 macro options.

Figure 1. Absolute Value Generator



IP Core Generator: Accumulator

Features

- **Accessible from the Macro Generator Dialog and HDLPlanner™ – Included in IDS for FPGA Devices and System Designer™ for AT94K FPSLIC™ Devices**
- **Variable Pitch Option**
- **Variable Width for Input and Output Devices**
- **Optional Set or Reset**
- **Optional Preset**
- **Clock Inversion Capability**
- **Initialization Polarity Selection**

Description

The Accumulator adds a given number to the register initial value. The functional description of the accumulator is as follows(1):

```
always(@posedge CLK or negedge RST)
begin
    if(RST == 'b0)
        SUM = 0;
    else if (ACC)
        {COUT, SUM} = SUM + DATA + CIN;
end
```

Note: 1. The above assumes that positive-edge clock and active-low reset have been specified.

Programmable SLI
AT40K
AT40KAL
AT94K

Application Note

Rev. 2424B–FPSLI–01/16/02

Parameters

Parameter	Value	Explanation
Pitch	Integer ≥ 1	Spacing between input pins. A pitch of 2 means will result in 1 cell between input pins
Width	Integer > 1	Width of input and output vectors
Initialization	Reset	Registers can be reset to zero
	Set	Registers can be set to one
	None	Registers are reset automatically on power-up
Preset	Value	Registers can be asynchronously loaded with a constant value
Invert Clock	Boolean	Invert the clock input
Initialization Polarity = Low	Boolean	Set/Reset/Preset input is active low
Preset Value Radix	Binary	Constants for preset are specified in binary representation
	Octal	Constants are specified in octal
	Decimal	Constants are specified in decimal
	Hex	Constants are specified in hexadecimal

Pins

Type	Name	Option	Explanation
In	CIN	No	Carry in
In	ACCUMULATE	No	Enables the accumulator, active high
In	DATA[Width - 1:0]	No	Data input
In	CL/CLKN	Yes	Clock (noninverted/inverted)
In	R/RN/S/SN	Yes	Reset/Set (active high/low)
Out	SUM[Width - 1:0]	No	Accumulator output
Out	COUT	No	Carry out(1)

Note: 1. Carry Out = SUM[Width - 1:0] + DATA[Width - 1:0] + CIN > 2^n - 1

Truth Table

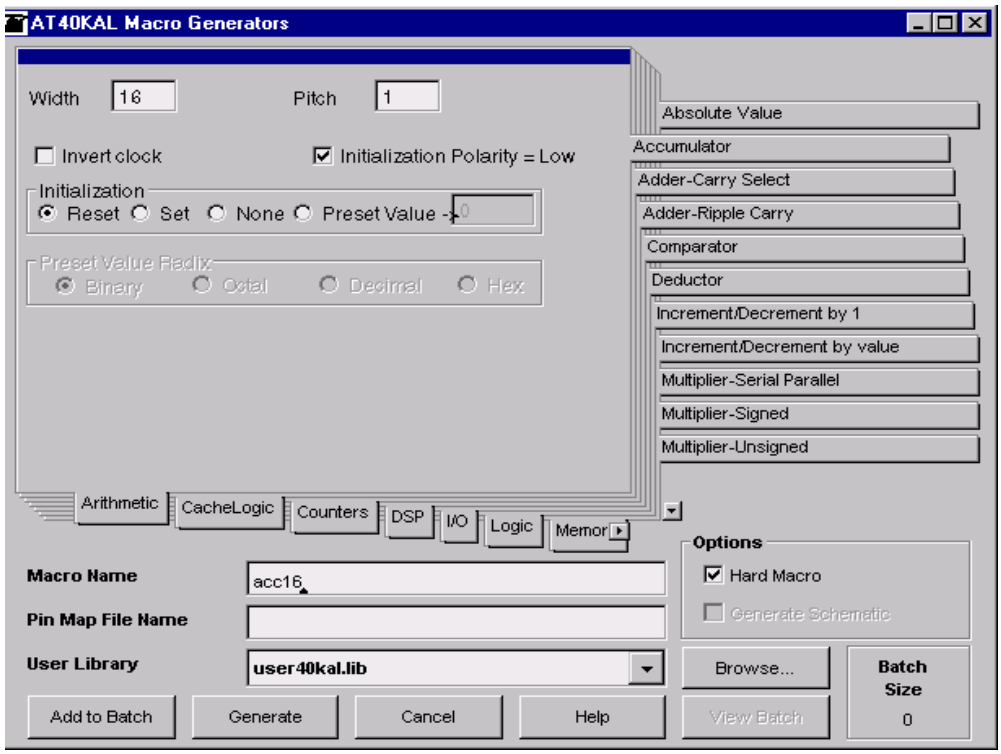
Input		Output	
CIN	DATA [W - 1:0]	SUM[W - 1:0]	COUT
A	B	A + B + SUM[W - 1:0]	1 if A + B + SUM[W - 1:0] > (2^W) - 1, 0 otherwise

Statistics

Device	Name	Speed (MHz)	Delay (ns)	Cells	Size (x * y)
AT40K	acc16	29.7	33.7	36	2 x 18
AT40K	acc8	48.8	20.5	20	2 x 10
AT94K/ AT40KAL	acc16	38.7	25.8	36	2 x 18
AT94K/ AT40KAL	acc8	61.6	16.2	20	2 x 10

Figure 1 shows an example of the acc16 macro options.

Figure 1. Accumulator Generator



IP Core Generator: Adders

Features

- **Adder – Carry Select**
- **Adder – Ripple Carry**
- **Accessible from the Macro Generator Dialog and HDLPlanner™ – Included in IDS for FPGA Devices and System Designer™ for AT94K FPSLIC™ Devices**
- **Variable Width for Input and Output Vectors**
- **Optional Carry In**
- **Optional Carry Out**
- **Ripple Carry Adder Only**
  - **Optional Registered Inputs and Outputs**
  - **Optional Signed Overflow Pin**
  - **Variable Pitch**
  - **Variable Width**
  - **Variable Aspect Ratio**

Adder – Carry Select

This generator can be used to generate an *n* bit Carry Select Adder.

Parameters

Parameter	Value	Explanation
Width	Integer > 1	Width of input and output vectors
Carry In	Boolean	Provide a carry-in pin
Carry Out	Boolean	Provide a carry-out pin

Pins

Type	Name	Option	Explanation
In	CIN	Yes	Carry in
In	DATAA[Width - 1:0]	No	A input
In	DATAB[Width - 1:0]	No	B input
Out	SUM[Width - 1:0]	No	Adder output
Out	COUT	Yes	Carry out

Programmable SLI
AT40K
AT40KAL
AT94K

Application Note

Rev. 2425B–FPSLI–01/17/02

Truth Table

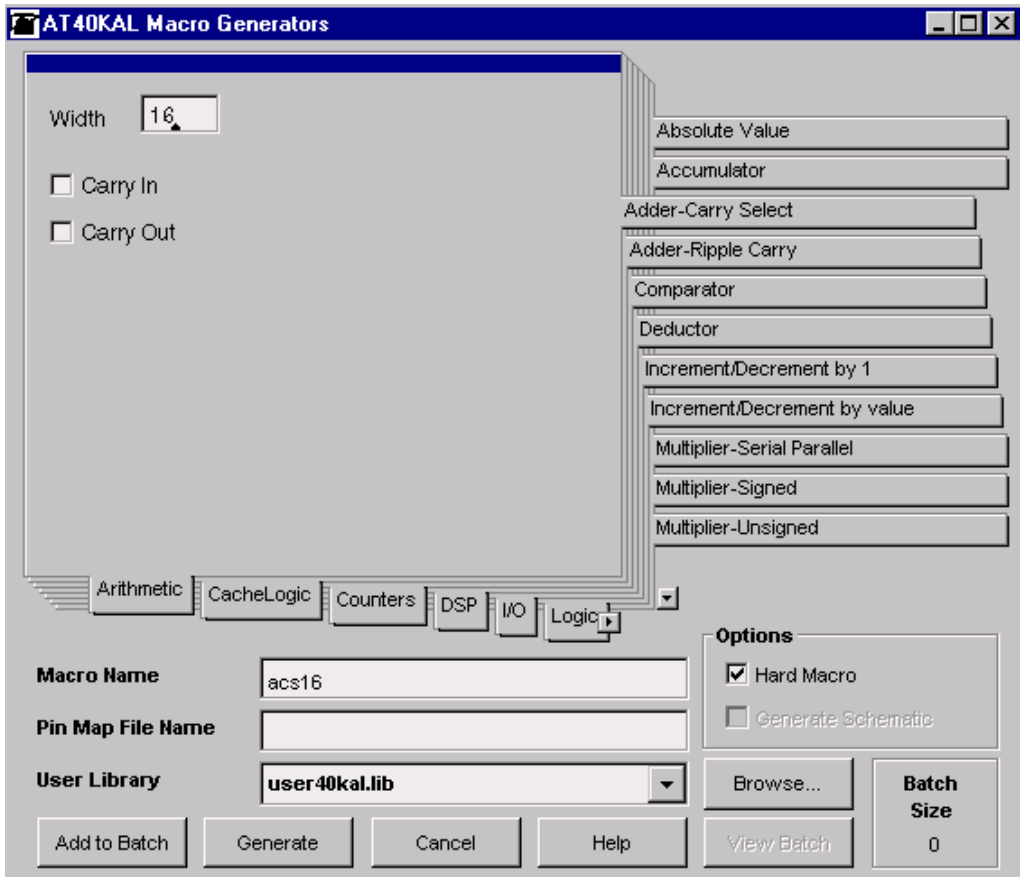
Input			Output	
CIN	DATAA[W - 1:0]	DATAB[W - 1:0]	SUM[W - 1:0]	COUT
C	A	B	A + B + C	1 if A + B + C > 2^W, 0 otherwise

Statistics

Device	Name	Speed (MHz)	Delay (ns)	Cells	Size (x * y)
AT40K	acs16	40.0	25.0	48	3 x 19
AT94K/ AT40KAL	acs16	47.0	21.3	48	3 x 19

Figure 1 shows an example of the acs16 macro options.

Figure 1. Adder – Carry Select Generator



Adder – Ripple Carry

The Adder generator can be used to generate a ripple carry adder.

Parameters

Parameter	Value	Explanation
Carry In	NoRegister	Include the carry in pin on the generator but do not register it
	Register	Register carry in of the adder
	Disabled	Do not include the carry-in pin on the adder
Carry Out	NoRegister	Include the carry-out pin on the adder but do not register it
	Register	Register carry out of the adder
	Disabled	Do not include the carry-out pin on the adder
Register	None	Do not register the inputs and outputs
	Input	Register inputs on the adder, exclude carry-in pin
	Output	Register outputs on the adder, exclude carry-out pin
	Both	Register both inputs and outputs including the carry-in and carry-out pins of the adder
Signed Over-flow Pin	Boolean	Provide a signed overflow output (treating input vectors as signed values)
Pitch	Integer > 1	Spacing between input pins, pitch of 2 means one cell between input pins
Width	Integer > 1	Width of input and output vectors
Aspect Ratio	Float ≥ 0.0	Aspect ratio of the adder layout. A ratio of 0.0 gives a thin, vertical layout, whereas a ratio of 1.0 gives a square layout

If input, output, carry-in or carry-out registers are selected, three additional parameters are available.

Register Parameters

Parameter	Value	Explanation
Invert Clock	Boolean	Invert the register clock
Initialization Polarity = Low	Boolean	Make register initialization active low
Register Set/Reset Function	Reset	Registers can be reset to zero
	Set	Registers can be set to one

Pins

Type	Name	Option	Explanation
In	CIN	Yes	Carry in
In	DATAA[Width - 1:0]	No	A input
In	DATAB[Width - 1:0]	No	B input
In	CLK/CLKN	Yes	Clock (noninverted/inverted)
In	R/RN/S/SN	Yes	Reset/Set (active high/low)
Out	SUM[Width - 1:0]	No	Adder output
Out	COUT	Yes	Carry out (cannot be used with overflow in an unsigned adder)
Out	OVERFLOW	Yes	Overflow

Carry out = DATAA + DATAB + CIN > 2^n - 1 or

DATAA + DATAB + CIN < -2^n

Signed overflow = DATAA + DATAB + CIN > 2^n - 1 or

DATAA + DATAB + CIN < -2^n

Truth Table

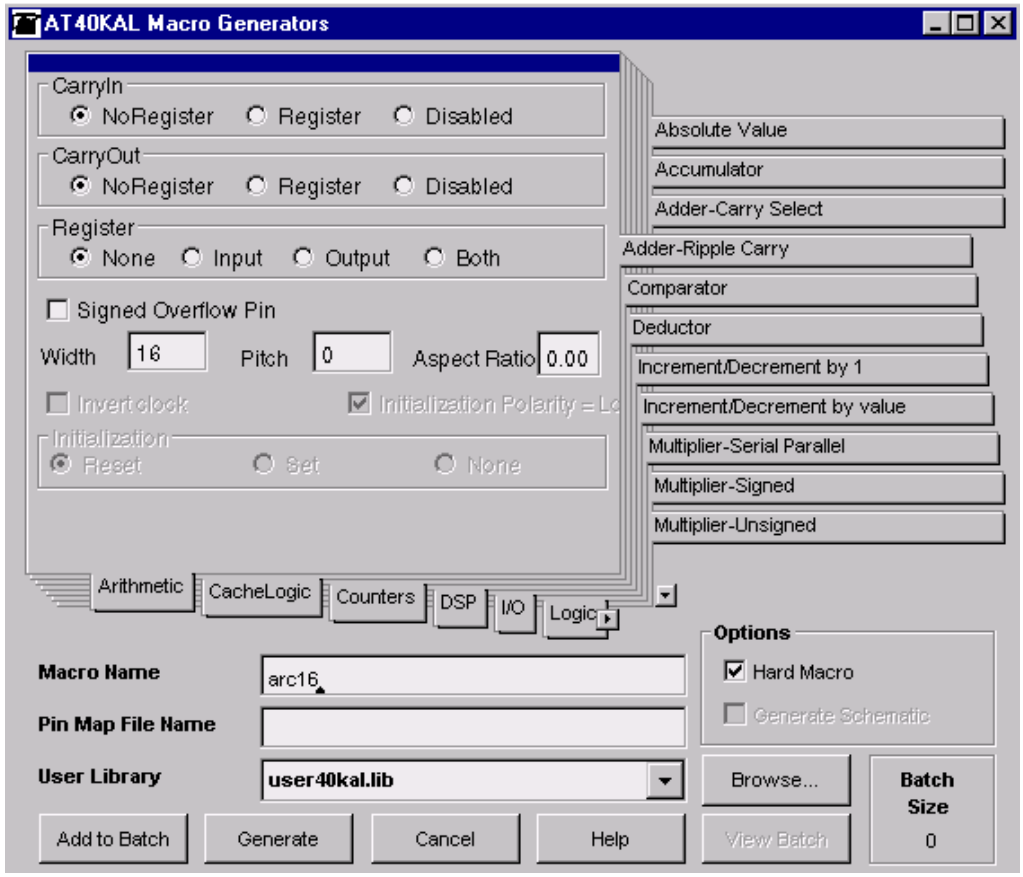
Input			Output	
CIN	DATAA[W - 1:0]	DATAB[W - 1:0]	SUM[W - 1:0]	COUT
C	A	B	A + B + C	1 if A + B + C > 2^W, 0 otherwise

Statistics

Device	Name	Speed (MHz)	Delay (ns)	Cells	Size (x * y)
AT40K	arc16	36.2	27.7	16	1 x 16
AT40K	arc8	69.2	14.5	8	1 x 8
AT94K/ AT40KAL	arc16	49.9	20.1	16	1 x 16
AT94K/ AT40KAL	arc8	95.6	10.5	8	1 x 8

Figure 2 shows an example of the arc16 macro options.

Figure 2. Adder – Ripple Carry Generator



IP Core Generator: I/O Buffer

Features

- **Bi-directional I/O Buffer**
- **Input I/O Buffer**
- **Output I/O Buffer**
- **Accessible from the Macro Generator Dialog and HDLPlanner™ – Included in IDS for FPGA Devices and System Designer™ for AT94K FPSLIC™ Devices**
- **Select TTL or CMOS Threshold on Inputs**
- **Optional Extra Delay on Inputs**
- **Optional Schmitt Trigger on Inputs**
- **Variable Slew Rate on Outputs**
- **Select Enable, Open Source or Open Drain on Outputs**
- **Optional Pull-up or Pull-down Resistors**
- **Variable Width**
- **Used Only with Schematic Designs**

Bi-directional I/O Buffer

The bi-directional I/O buffer generator can be used to generate a soft macro (schematic only) which uses the specified options. This macro is not stored in the library, but becomes a part of the design. It generates a schematic symbol that can be used for blocks of I/O to simplify schematics. The generator provides a simple means for connecting I/Os to buses within the design. It also facilitates I/O selection as all of the parameters can be specified and the program will choose the appropriate cell from the library.

Programmable SLI
AT40K
AT40KAL
AT94K

Application Note

Rev. 2426B–1/02

Parameters

Parameter	Value	Explanation
Input Threshold	TTL	The input threshold is TTL compatible
	CMOS	The input threshold is CMOS compatible
Input Extra Delay (ns)	0	The input has no additional delay associated with it
	1	The input has an extra delay of approximately 1 ns
	3	The input has an extra delay of approximately 3 ns
	5	The input has an extra delay of approximately 5 ns
Input Schmitt Triggering	Boolean	The input Schmitt trigger circuit is enabled
Output Slewrate	Fast	The output buffer has fast drive (maximum slew rate)
	Medium	The output buffer has medium drive (medium slew rate)
	Slow	The output buffer has standard drive (reduced slew rate)
Output Type	Enable	The output has an enable pin
	Open Source	The output is an open source
	Open Drain	The output is an open drain
Pull Resistor	None	Pad pins have no pull-up or pull-down resistor
	Pull Up	Pad pins have pull-up resistor
	Pull Down	Pad pins have pull-down resistor
Width	Integer > 0	Width of input and output data

Pins

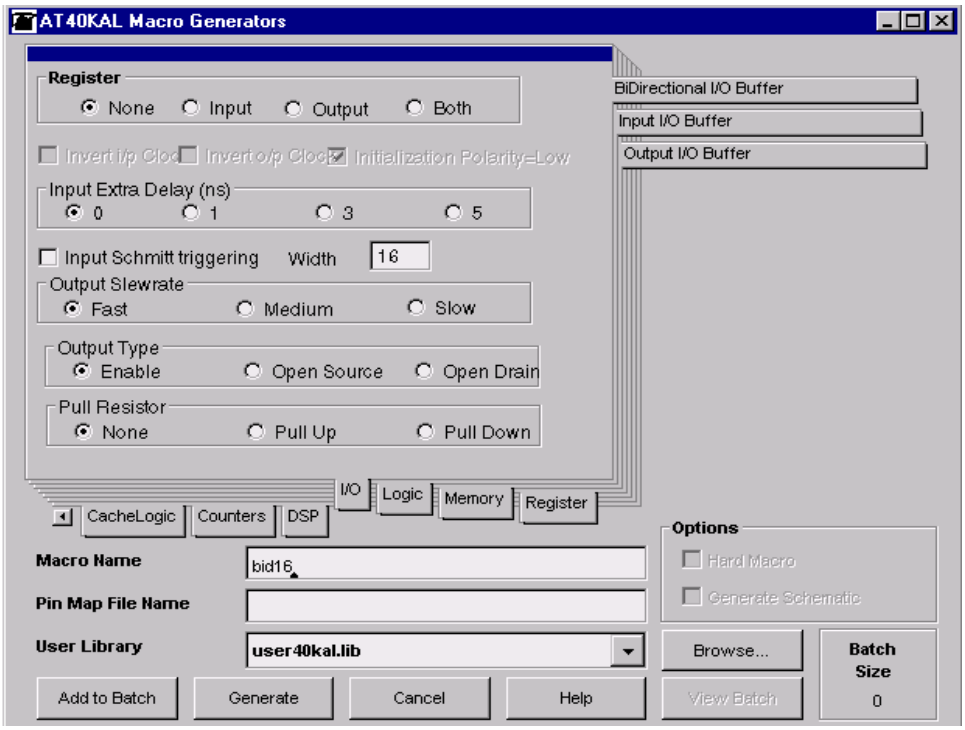
Type	Name	Option	Explanation
In	A[Width - 1:0]	No	Data input from the core to the I/O
In	OE[Width - 1:0]	Yes	Output enable input from the core to the I/O
In/Out	PAD[Width - 1:0]	No	Pad pin of I/O (bi-directional)
Out	Q[Width - 1:0]	No	Output from the I/O to the core

The following user configurable memory bits, which are set depending on the previous page selection of parameters, provide control of the I/O logic.

- TTL/CMOS Inputs: A user configurable bit determining the threshold level (TTL or CMOS) of the input buffer.
- Schmitt Triggering: A user configurable bit determining whether a Schmitt trigger circuit on the input pad should be enabled or disabled. The Schmitt trigger is a regenerative comparator circuit, which improves the rise and fall times (leading and trailing edges) of the incoming signal.
- Extra Delay: The input buffer can have four different intrinsic delays. This lets the user specify an extra delay on the input signal in order to meet any data-held requirements. A value of “0” means no extra delay above the intrinsic delay of the input buffer. Delays of approximately 1, 3 and 5 ns can also be set.
- Open Source/Open Drain/Tri-state Outputs: User configurable bits that set the output drive to either tri-state, open source (1 or Z) or open drain (0 or Z).
- Slew Rate Control: User configurable bits that control the output drive. When set to “FAST”, the output buffer has full drive capability 20-mA buffer. The “MEDIUM” setting produces a medium-drive 14-mA buffer, while “SLOW” gives a standard-drive 6-mA buffer.
- Pull-up/Pull-down: User configurable bits controlling the pull-up and pull-down transistors in the I/O pin. These supply either a weak “1” or a weak “0” level to the pad pin. When all other drivers are off, this control will dictate the signal level of the pad pin.

Figure 1 shows an example of the bid16 macro options.

Figure 1. Bi-directional I/O Buffer Generator



Input I/O Buffer

The input I/O buffer generator can be used to generate a soft macro (schematic only) which uses the specified options. This macro is not stored in the library, but becomes a part of the design. The generator provides a simple means for connecting I/Os to buses within the design. It also facilitates I/O selection as all of the parameters can be specified and the program will choose the appropriate cell from the library.

Parameters

Parameter	Value	Explanation
Threshold	TTL	Input threshold is TTL compatible
	CMOS	Input threshold is CMOS compatible
Width	Integer > 0	Width of input and output data
Extra Delay	0	The input has no additional delay associated with it
	1	The input has an extra delay of approximately 1 ns
	3	The input has an extra delay of approximately 3 ns
	5	The input has an extra delay of approximately 5 ns
Pull Resistor	None	Pad pins have no pull-up or pull-down resistor
	Pull-up	Pad pins have pull-up resistor
	Pull-down	Pad pins have pull-down resistor
Schmitt Triggering	Boolean	The input Schmitt trigger circuit is enabled

Pins

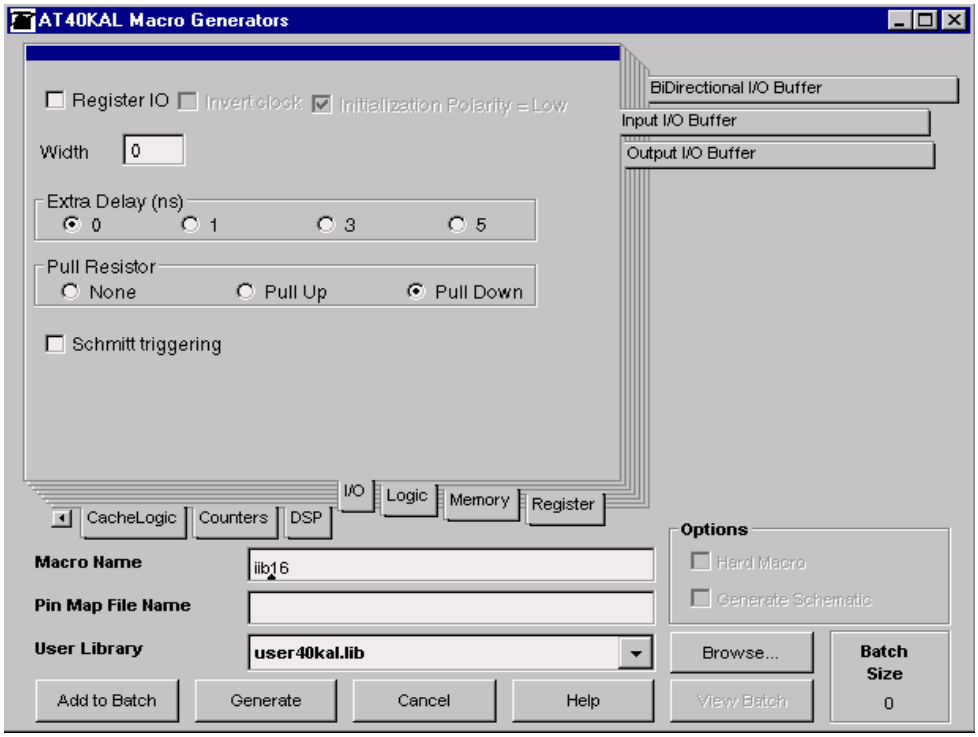
Type	Name	Option	Explanation
In	PAD[Width - 1:0]	No	Data input to the chip (pad pin)
Out	Q[Width - 1:0]	No	Output from the I/O to the core

The following user configurable memory bits, which are set depending on the above selection of parameters, provide control of the I/O logic.

- TTL/CMOS Inputs: A user configurable bit determining the threshold level (TTL or CMOS) of the input buffer.
- Schmitt Triggering: A user configurable bit determining whether a Schmitt trigger circuit on the input pad should be enabled or disabled. The Schmitt trigger is a regenerative comparator circuit, which improves the rise and fall times (leading and trailing edges) of the incoming signal.
- Extra Delay: The input buffer can have four different intrinsic delays. This lets the user specify an extra delay on the input signal in order to meet any data hold requirements. A value of “0” means no extra delay above the intrinsic delay of the input buffer. Delays of approximately 1, 3 and 5 ns can also be set.
- Pull-up/Pull-down: User configurable bits controlling the pull-up and pull-down transistors in the I/O pin. These supply either a weak “1” or a weak “0” level to the pad pin. When all other drivers are off, this control will dictate the signal level of the pad pin.

Figure 2 shows an example of the iib16 macro options.

Figure 2. Input I/O Buffer Generator



Output I/O Buffer

The output I/O buffer generator can be used to generate a soft macro (schematic only) which uses the specified options. This macro is not stored in the library, but becomes a part of the design. The generator provides a simple means for connecting I/Os to buses within the design. It also facilitates I/O selection as all of the parameters can be specified and the program will choose the appropriate cell from the library.

Parameters

Parameter	Value	Explanation
Type	Normal	The output has no enable pin
	Enable	The output has an enable pin
	Open Source	The output is an open source
	Open Drain	The output is an open drain
Width	Integer > 0	Width of output data
Slewrate	Fast	The output buffer should be fast drive (maximum slew rate)
	Medium	The output buffer should be medium drive (medium slew rate)
	Slow	The output buffer should be standard drive (reduced slew rate)
Pull Resistor	None	Pad pins have no pull-up or pull-down resistor
	Pull-up	Pad pins have pull-up resistor
	Pull-down	Pad pins have pull-down resistor

Pins

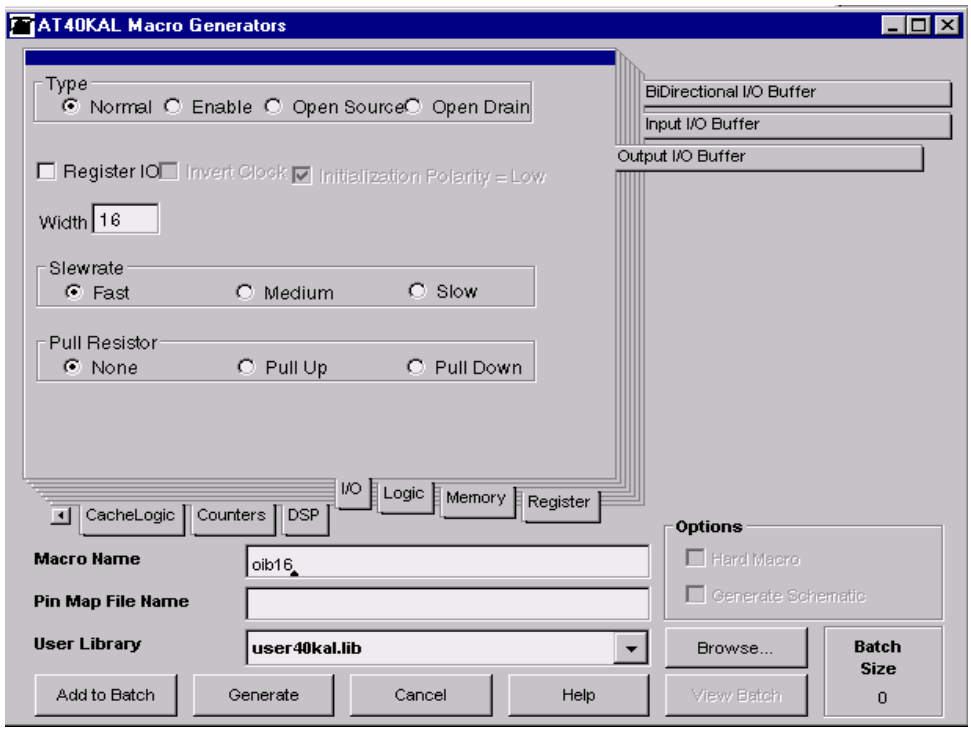
Type	Name	Option	Explanation
In	A[Width - 1:0]	No	Data input from the chip
In	OE[Width - 1:0]	Yes	Tri-state enable pins
Out	PAD[Width - 1:0]	No	Output from the I/O (pad pin)

The following user configurable memory bits, which are set depending on the above selection of parameters, provide control of the I/O logic.

- Open Source/Open drain/Tri-state Outputs: User configurable bits that set the output drive to either tri-state, open source (1 or Z) or open drain (0 or Z).
- Slew Rate Control: User configurable bits that control the output drive. When set to “FAST”, the output buffer has full drive capability 20-mA buffer. The “MEDIUM” setting produces a medium-drive 14-mA buffer, while “SLOW” gives a standard-drive 6-mA buffer.
- Pull-up/Pull-down: User configurable bits controlling the pull-up and pull-down transistors in the I/O pin. These supply either a weak “1” or a weak “0” level to the pad pin. When all other drivers are off, this control will dictate the signal level of the pad pin.

Figure 3 shows an example of the oib16 macro options.

Figure 3. Output I/O Buffer Generator



IP Core Generator: Bus Ripper

Features

- **Accessible from the Macro Generator Dialog and HDLPlanner™ – Included in IDS for FPGA Devices and System Designer™ for AT94K FPSLIC™ Devices**
- **Variable Width of Input Bus**
- **Used for Schematic Designs Only**

Bus Ripper

The Bus Ripper generator allows the user to produce variable-width “soft” buffers for re-naming buses within a design. When the design is imported into IDS, the mapping tools will identify the bus ripper macro as redundant logic and remove it from the netlist. The bus rippers are a fast and convenient method of re-labeling buses in a design without consuming unnecessary logic resources in the final implementation.

Note: Ensure that Mapping is enabled when importing a circuit containing bus ripper macros, to prevent extra logic from being added to the design. From the **Options** menu, select **Options** and then **Mapping**. The **Mapping Enabled** option should be checked.

Parameters

Parameter	Value	Explanation
Width	Integer	Width of input bus

Pins

Type	Name	Option	Explanation
In	INPUT[Width - 1:0]	No	Input bus
In	OUTPUT[Width - 1:0]	No	Output bus

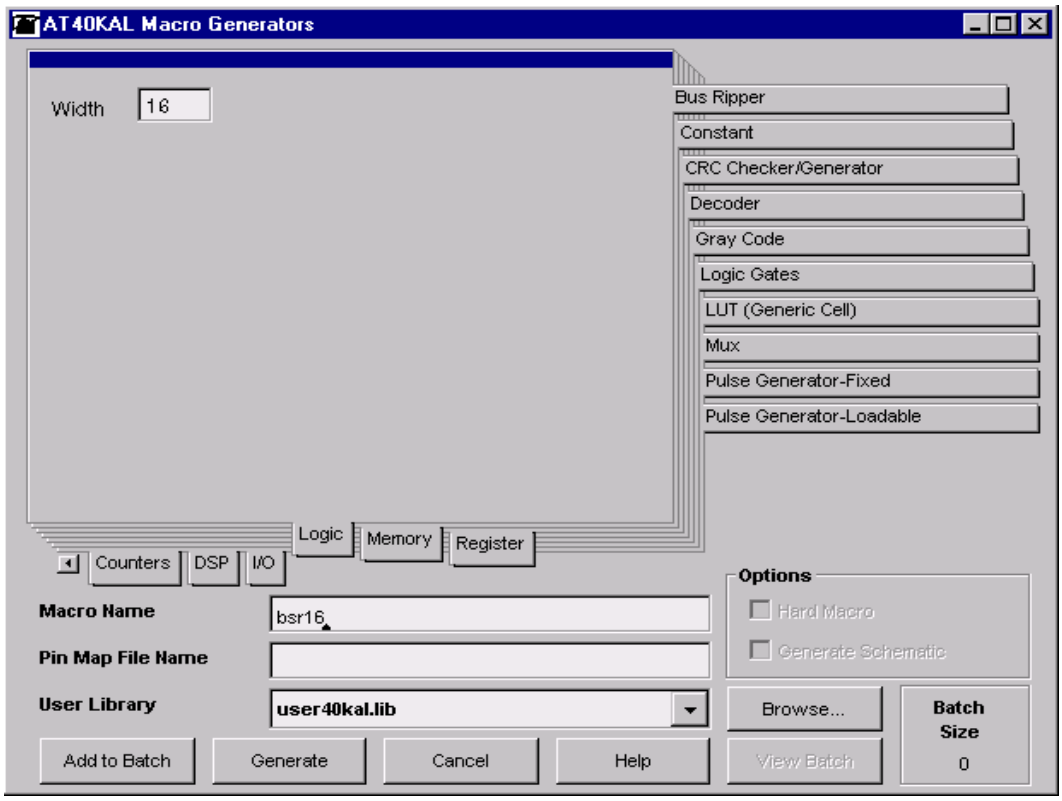
Figure 1 shows an example of the bsr16 macro options.

Programmable SLI
AT40K
AT40KAL
AT94K

Application Note

Rev. 2427B–1/02

Figure 1. Bus Ripper Generator



IP Core Generator: Comparator

Features

- **Accessible from the Macro Generator Dialog and HDLPlanner™ – Included in IDS for FPGA Devices and System Designer™ for AT94K FPSLIC™ Devices**
- **Optional Signed or Unsigned Inputs**
- **Variable Width of Inputs A and B**
- **Variable Outputs**

Comparator

The function of the Comparator generator can be explained as follows:

Equals	= (DATAA == DATAB)
NotEquals	= (DATAA != DATAB)
LessThan	= (DATAA < DATAB)
LessThanOrEqualTo	= (DATAA ≤ DATAB)
GreaterThan	= (DATAA > DATAB)
GreaterThanOrEqualTo	= (DATAA ≥ DATAB)

Parameters

Parameter	Value	Explanation
Representation	Unsigned	Treat inputs as unsigned
	Signed	Treat inputs as signed
Width	Integer > 1	Width of inputs A and B
A = B[1]	Boolean	Add Equal pin (AEB) to comparator
A != B	Boolean	Add NotEqual pin (ANEB) to comparator
A < B	Boolean	Add LessThan pin (ALB) to comparator
A ≤ B	Boolean	Add LessThanOrEqualTo pin (ALEB) to comparator
A > B	Boolean	Add GreaterThan pin (AGB) to comparator
A ≥ B	Boolean	Add GreaterThanOrEqualTo pin (AGEB) to comparator

Note: 1. If A = B is the only output selected, an optimized Equality-Only comparator will be produced, resulting in a much smaller layout.

Programmable SLI
AT40K
AT40KAL
AT94K

Application Note

Rev. 2428B–FPSLI–01/16/02

Pins

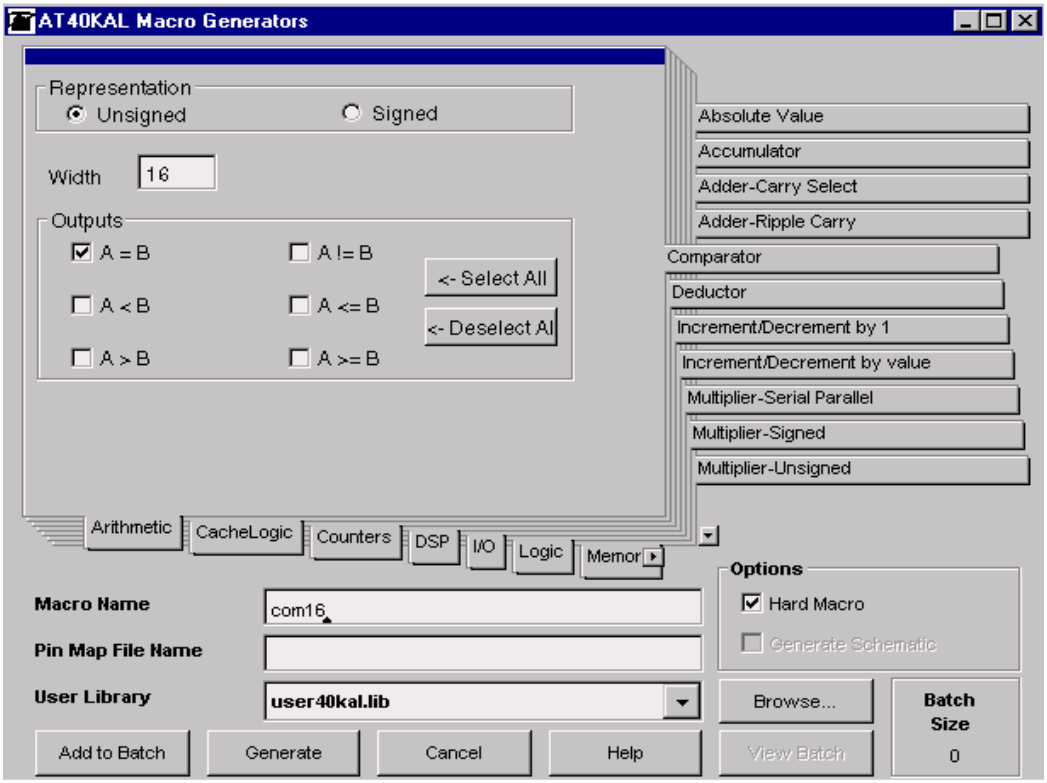
Type	Name	Option	Explanation
In	DATAA[Width - 1:0]	No	Input data A
In	DATAB[Width - 1:0]	No	Input data B
Out	AEB	Yes	1 if A = B, 0 otherwise
Out	ANEB	Yes	1 if A != B, 0 otherwise
Out	ALB	Yes	1 if A < B, 0 otherwise
Out	ALEB	Yes	1 if A ≤ B, 0 otherwise
Out	AGB	Yes	1 if A > B, 0 otherwise
Out	AGEB	Yes	1 if A ≥ B, 0 otherwise

Statistics

Device	Name	Speed (MHz)	Delay (ns)	Cells	Size (x * y)
AT40K	com16	45.4	22	16	8 x 2
AT40K	com8	97.4	10.3	8	4 x 2
AT40KAL/ AT94KAL	com16	56.1	17.8	16	8 x 2
AT40KAL/ AT94KAL	com8	128.0	7.8	8	4 x 2

Figure 1 shows an example of the com16 macro options.

Figure 1. Comparator Generator



IP Core Generator: Constant

Features

- **Accessible from the Macro Generator Dialog and HDLPlanner™ – Included in IDS for FPGA Devices and System Designer™ for AT94K FPSLIC™ Devices**
- **Variable Width of Output Vectors**
- **Constant Value**
- **Binary, Octal, Decimal or Hexadecimal Radix Value**

Constant

A Constant can be generated with the specified values. Note that this generator has to be run with the “Hard Macro” option deselected, since constants should be implemented as soft macros so that the mapper can “fold” them in with other logic in the design.

Parameters

Parameter	Value	Explanation
Width	Integer > 0	Width of output vector
Constant Value	Integer ≥ 0	Value of the constant
Radix	Binary	Constant value is specified using binary representation
	Octal	Value is octal
	Decimal	Value is decimal
	Hex	Value is hexadecimal

Pins

Type	Name	Option	Explanation
Out	RESULT[Width - 1:0]	No	Constant value, if Width is not large enough, least significant bits of value will be used

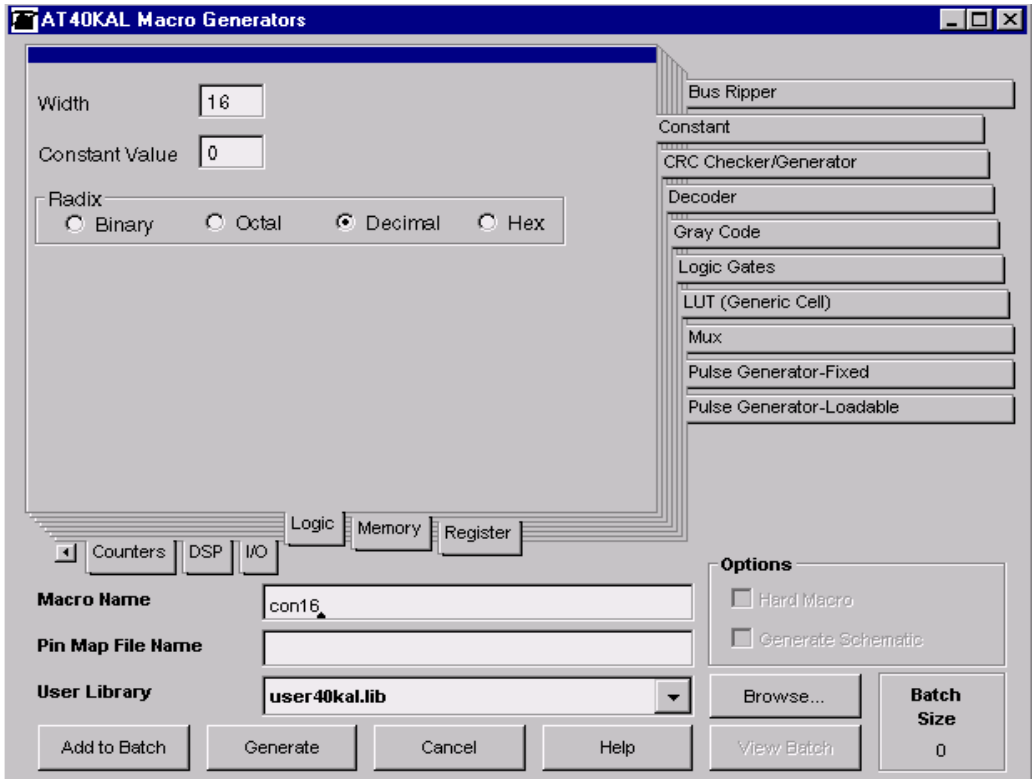
Figure 1 shows an example of the con16 macro options.

Programmable SLI
AT40K
AT40KAL
AT94K

Application Note

Rev. 2429B–1/02

Figure 1. Constant Generator



IP Core Generator: Counter

Features

- Counter – Johnson
- Counter – LFSR
- Counter – PreScaled
- Counter – Ripple Carry
- Counter – Terminal
- Accessible from the Macro Generator Dialog and HDLPlanner™ – Included in IDS for FPGA Devices and System Designer™ for AT94K FPSLIC™ Devices
- Clock Inversion Capability
- Initialization Polarity Selection
- Counter Johnson Only
  - Variable Width of Output Vectors
  - Optional Enable
  - Optional Fold Layout
  - Registers Initialization Selection
- Counter LFSR Only
  - Generator Output Selection
  - Variable Width of Input and Output Data
  - Variable Pitch of Output Pins
  - Optional Fold Layout
  - Count Sequence Report File
  - Count Sequence Formatting File
  - Registers Initialization Selection
- Counter PreScaled Only
  - Counter Direction Selection
  - Variable Width of Output Vectors
  - Variable Pitch of Output Pins
  - Parallel Load Capability
  - Registers Initialization Selection
  - Preset Value Radix Selection
- Counter Ripple Carry Only
  - Counter Direction Selection
  - Variable Width of Output Vector
  - Optional Enable
  - Optional Fold Layout
  - Parallel Load Capability Selection
  - Registers Initialization Selection
  - Preset and Load Value Radix Selection
- Counter Terminal Only
  - Terminal Count Capability
  - Radix of Terminal Selection
  - Variable Pitch of Output Pins

Programmable SLI

AT40K

AT40KAL

AT94K

Application Note

Rev. 2430B–FPSLI–01/16/02

Counter – Johnson

The Johnson Counter generator can be used to generate counters in which only one output changes on each clock cycle. The following parameters are available:

Parameters

Parameter	Value	Explanation
Width	Integer > 1	Width of output vector
Enable	Boolean	Add an Enable pin to component
Fold	Boolean	Fold layout in half
Invert Clock	Boolean	Invert the clock input
Initialization Polarity = Low	Boolean	Set/Reset/Preset input is active low
Initialization	Reset	Registers can be reset to zero
	Set	Registers can be set to one
	None	Registers are initialized automatically on power-up

Pins

Type	Name	Option	Explanation
In	R/RN/S/SN	No	Reset/Set (active high/low)
In	CLK/CLKN	No	Clock (noninverted/inverted)
In	ENABLE	Yes	Enable counter
Out	Q[Width - 1:0]	No	Counter output

Truth Table(1)

Input			Output
RN	CLK	ENABLE	Q[W - 1:0]
0	X	X	0
1	X	0	Q[W - 1:0]
1	R	1	Q[W - 2:0]Q[W -1]

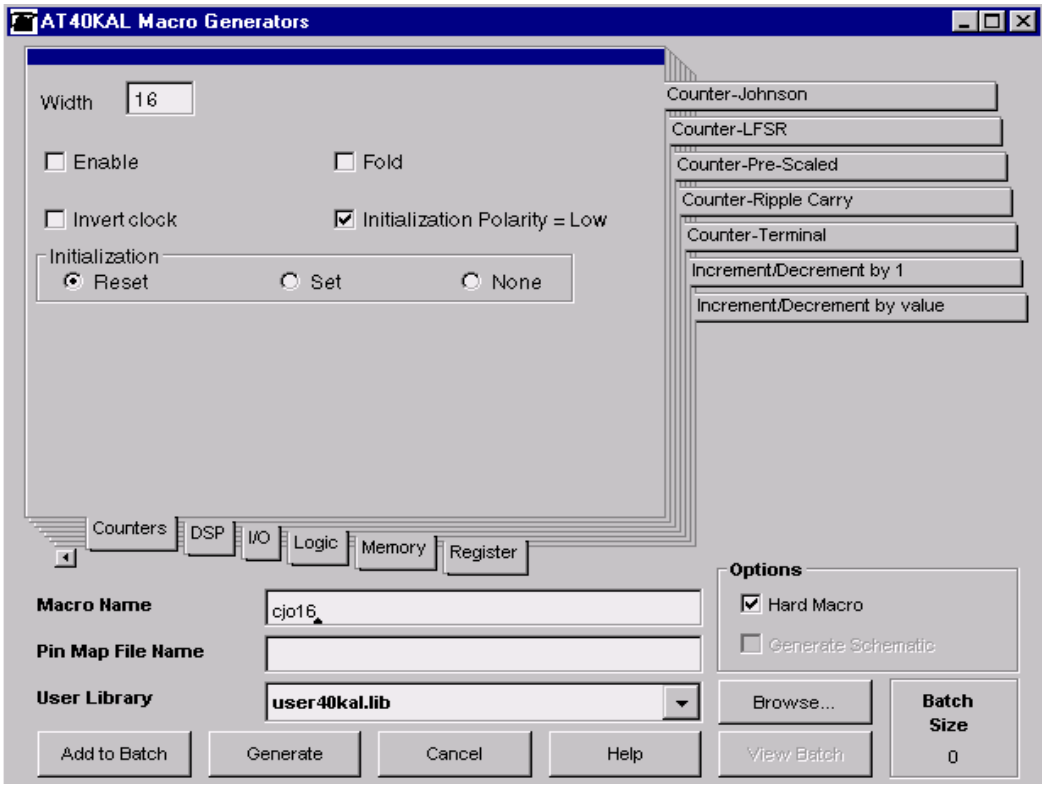
Note: 1. This truth table assumes that a noninverted clock and active low reset have been selected.

Statistics

Device	Name	Speed (MHz)	Delay (ns)	Cells	Size (x * y)
AT40K	cjo16	95.4	10.5	16	1 x 16
AT40K	cjo8	124.1	8.1	8	1 x 8
AT94K/ AT40KAL	cjo16	98.0	10.2	16	1 x 16
AT94K/ AT40KAL	cjo8	120.2	8.3	8	1 x 8

Figure 1 shows an example of the cjo16 macro options.

Figure 1. Counter – Johnson Generator



Counter – LFSR

The LFSR generator can be used to create a high-speed divide by [2**(n - 1) -1] counter, where *n* is the number of bits. The count sequence can be output in a format specified by a user-supplied data file (described below).

Parameters

Parameter	Value	Explanation
Generator Output	Component	Generate component only – do not output count sequence
	Count File	Output count sequence only – do not generate component
	Both	Output count sequence and generate component
Width	Integer ≥ 3	Width of input and output data
Pitch	Integer ≥ 1	Pitch of output pins. A pitch of 2 results in a one cell gap between outputs
Fold Layout	Boolean	Fold layout in half
Count Sequence Report File	Filename	Name of count sequence output file. This file is written out to the current project directory when “Count file” or “Both” are selected as the generator output option.
Count Sequence Formatting File	Filename	Name of the file (found in the current project directory) that is to be used to format the count sequence output.
Invert Clock	Boolean	Invert the clock input
Initialization Polarity = Low	Boolean	Set/Reset/Preset input is active low
Initialization	Reset	Registers can be reset to zero
	Set	Registers can be set to one
	None	Registers are initialized automatically on power-up
	Preset	Registers can be asynchronously loaded with a constant value
Preset Value Radix	Binary	Preset value is specified in binary representation
	Octal	Preset value is specified in octal representation
	Decimal	Preset value is specified in decimal representation
	Hex	Preset value is specified in hexadecimal representation

Pins

Type	Name	Option	Explanation
In	R/RN/S/SN/P/PN	No	Reset/Set/Preset (active high/low)
In	CLK/CLKN	No	Clock (noninverted/inverted)
Out	Q[Width - 1:0]	No	Counter output

Truth Table[1]

Input		Output
RN	CLK	Q[W - 1:0]
0	X	0
1	X	Present State
1	R	LFSR Next State

Note: 1. This truth table assumes a noninverted clock and active low reset have been selected.

Count Sequence Output – When the Generator Output option is set to Count or Both, an output file called the Count sequence report file is generated on completion of the program run. This file lists the count sequence for the generated LFSR.

The user can also create an input file, called the Count sequence formatting file, to specify which parts of the count sequence should be listed. The Count sequence formatting file should be placed in the design directory. The format of the file is basically a header line with “hex”, “dec” or “bin” to indicate that the numbers in the file are in hexadecimal, decimal or binary format. Each subsequent line should be in the form of Start_Count End_Count. For example:

lfsr.dat

dec

0 2

4 20

88 127

300 844

1088 1090

The output will have the format Count *count_number* Decode *decode_value*, where *decode_value* is the counter Q[W-1:0] output for *count_number*.

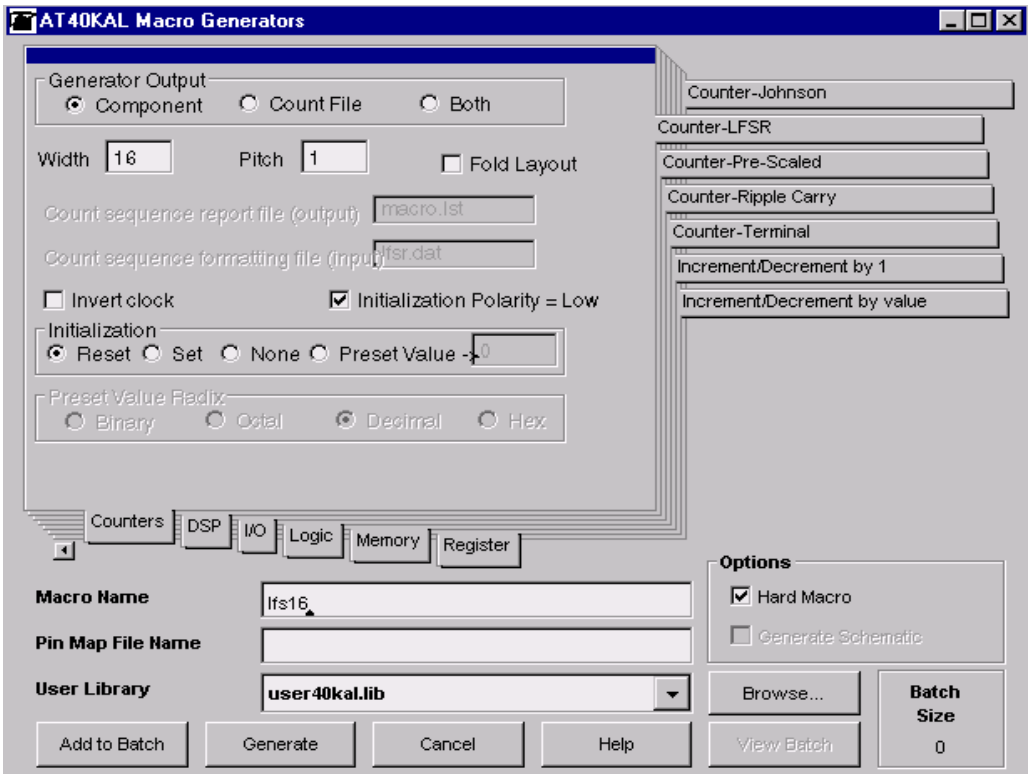
Decoding Terminal Count – The count sequence always ends with the same sequence: all bits are 1’s after which 0’s shift in on each clock from the LSB until all bits are 0’s. For example, a 3 bit counter would end with the sequence 7, 6, 4, 0. A simple comparator circuit can therefore be used to quickly decode the Terminal Count.

Statistics

Device	Name	Speed (MHz)	Delay (ns)	Cells	Size (x * y)
AT40K	lfs16	93.5	10.7	16	1 x 16
AT40K	lfs8	115.7	8.6	8	1 x 8
AT94K/ AT40KAL	lfs16	94.7	10.6	16	1 x 16
AT94K/ AT40KAL	lfs8	110.6	9.0	8	1 x 8

Figure 2 shows an example of the lfs16 macro options.

Figure 2. Counter – LFSR Generator



Counter – PreScaled

The PreScaled Counter generator can be used to implement a faster Ripple Carry binary counter.

Parameters

Parameter	Value	Explanation
Direction	Up	Create an up counter
	Down	Create a down counter
	Up/Down	Create a programmable up/down counter
Width	Integer > 2	Width of input and output vectors
Pitch	Integer ≥ 1	Spacing between output pins. A pitch of 2 means one cell between pins
Parallel Load	Boolean	Provide a parallel load capability
Enable	Boolean	Provide an enable input
Invert Clock	Boolean	Invert the clock input
Initialization Polarity = Low	Boolean	Set/Reset/Preset input is active low
Initialization	Reset	Registers can be reset to zero
	Set	Registers can be set to one
	None	Registers are reset automatically on power-up
	Preset	Registers can be asynchronously loaded with a constant value
Preset Value Radix	Binary	Preset value is specified in binary representation
	Octal	Preset value is specified in decimal representation
	Decimal	Preset value is specified in octal representation
	Hex	Preset value is specified in hexadecimal representation

Pins

Type	Name	Option	Explanation
In	SLOAD	Yes	Performs a parallel load when low
In	ENABLE	Yes	Enables the counter, active high
In	CARRYIN	No	Carry in to counter
In	UP_DOWN	No	Up = 1, down = 0
In	DATA[Width - 1:0]	Yes	Parallel load data input
In	CLK/CLKN	Yes	Clock (noninverted/inverted)
In	R/RN/S/SN/P/PN	Yes	Reset/Set/Preset (active high/low)
Out	Q[Width - 1:0]	No	Counter output
Out	RCO	No	Carry out(1)

Note: 1. If Up, then Carry out = 1 when counter overflows. Carry out = 1 when counter underflows.

Truth Table[1]

Input					Output	
SLOAD	ENABLE	UP_DOWN	CARRYIN	DATA [W - 1:0]	Q[W - 1:0]	RCO
0	X	1	X	A	A	1 if A > (2^W) - 1, 0 otherwise
0	X	0	X	A	A	1 if A < -(2^W), 0 otherwise
1	0	X	X	X	Present State	Present State
1	X	X	0	X	Q[W - 1:0]	RCO
1	1	1	1	X	Q[W - 1:0] + 1	1 if Q[W - 1:0] + 1 > (2^W) - 1, 0 otherwise
1	1	0	1	X	Q[W - 1:0] - 1	1 if Q[W - 1:0] - 1 < -(2^W), 0 otherwise

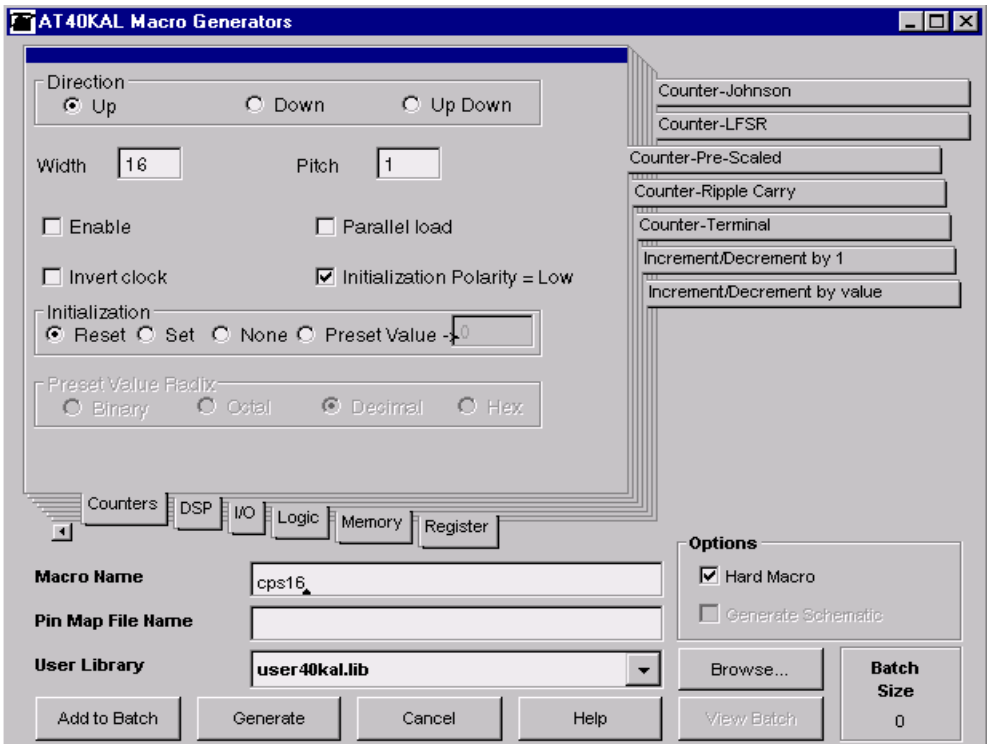
Note: 1. The ENABLE signal must be negated during load, and must remain negated for at least 1 clock cycle after SLOAD negates.

Statistics

Device	Name	Speed (MHz)	Delay (ns)	Cells	Size (x * y)
AT40K	cps16	36.1	27.7	17	1 x 17
AT40K	cps8	68.8	14.5	9	1 x 9
AT94K/ AT40KAL	cps16	45.2	22.1	17	1 x 17
AT94K/ AT40KAL	cps8	82.1	12.2	9	1 x 9

Figure 3 shows an example of the cps16 macro options.

Figure 3. Counter – PreScaled Generator



Counter – Ripple Carry

The Counter generator can be used to generate Ripple Carry Counters. The following parameters are available for the counters.

Parameters

Parameter	Value	Explanation
Direction	Up	Create an Up counter
	Down	Create a Down counter
	Up/Down	Create an Up/Down counter
Width	Integer > 1	Width of input and output vectors
Enable	Boolean	Add an enable pin to component
Fold	Boolean	Fold layout in half
Parallel load	Disabled	No parallel load capability
	Var. Value	Parallel load capability
	Const. Value	Fixed value can be synchronously loaded
Initialization	Reset	Registers can be reset to zero
	Set	Registers can be set to one
	None	Registers are initialized automatically on power-up
	Preset	Registers can be asynchronously loaded with a constant value
Invert Clock	Boolean	Invert the clock input
Initialization Polarity = Low	Boolean	Set/Reset/Preset input is active low
Preset & Load Value Radix	Binary	Constants for parallel load and preset are specified in binary representation
	Octal	Constants are specified in octal
	Decimal	Constants are specified in decimal
	Hex	Constants are specified in hexadecimal

Pins

Type	Name	Option	Explanation
In	R/RN/S/SN/P/PN	No	Reset/Set/Preset (active high/low)
In	CLK/CLKN	No	Clock (Noninverted/Inverted)
In	ENABLE	Yes	Enable counter
In	DATA[Width - 1:0]	Yes	Parallel load input
In	LOAD	Yes	Load signal (active low)
In	UP_DOWN	Yes	Up/Down control up = 1
Out	Q[Width - 1:0]	No	Counter output
Out	RCO	No	Ripple carry out

Truth Table[1]

Input						Output	
RN	ENABLE	CLK	LOAD	DATA[W - 1:0]	UP/DOWN	Q[W - 1:0]	RCO
0	X	X	X	X...X	X	0...0	0
1	0	X	1	X...X	X	QW...Q0	0
1	1	R	0	DW...D0	X	DW...D0	0
1	1	X	1	X...X	X	Present State	QW*...*Q1*Q0
1	1	R	1	DW...D0	1	DW...D0 + 1	QW*...*Q1*Q0
1	1	R	1	DW...D0	0	DW...D0 - 1	QW*...*Q1*Q0

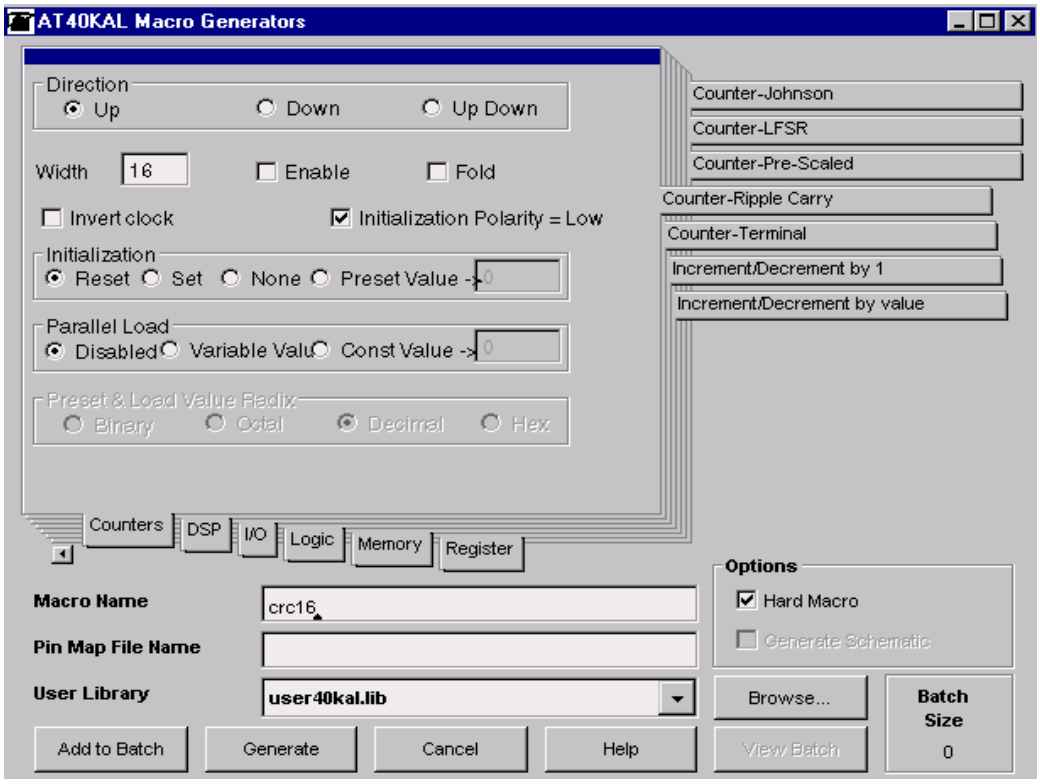
Note: 1. This truth table assumes that an active-low reset and non-inverted clock have been selected. For an up counter, the RCO pin goes high whenever the current state equals the maximum count. For a down counter, it goes high when the minimum count state is reached (i.e. all zeros). Counters can be chained together by connecting the RCO pin of one counter to the enable input of the next.

Statistics

Device	Name	Speed (MHz)	Delay (ns)	Cells	Size (x * y)
AT40K	crc16	36.8	27.2	16	1 x 16
AT40K	crc8	71.5	14.0	8	1 x 8
AT40KAL/ AT94KAL	crc16	45.7	21.9	16	1 x 16
AT40KAL/ AT94KAL	crc8	83.6	12.0	8	1 x 8

Figure 4 shows an example of the crc16 macro options.

Figure 4. Counter – Ripple Carry Generator



Counter – Terminal

The Terminal Counter generator allows the user to create a counter that stops after *n* clock cycles. The following options are available:

Parameters

Parameter	Value	Explanation
Terminal Count	Integer ≥ 1	The number of states the counter should step through before stopping and asserting the TERMCNT pin.
Radix of Terminal Count	Binary	Terminal count value is specified in binary representation
	Octal	Value is specified in octal
	Decimal	Constants are specified in decimal
	Hex	Constants are specified in hexadecimal
Pitch	Integer >= 1	Pitch between output pins. A pitch of 2 results in a one cell gap between pins.
Invert Clock	Boolean	Invert the clock input
Initialization Polarity = Low	Boolean	Preset input is active low

Pins

Type	Name	Option	Explanation
In	CLK/CLKN	No	Clock (noninverted/inverted)
In	P/PN	No	Preset input (active high/low)
Out	TERMCNT	No	Terminal count(1)

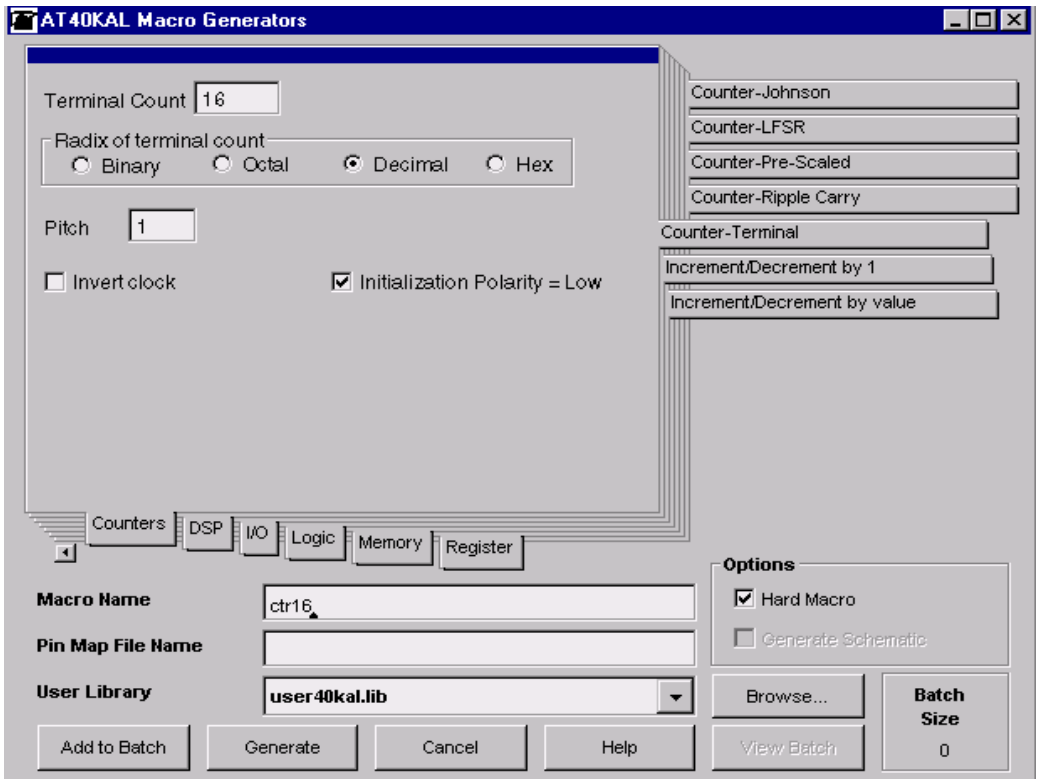
Note: 1. Terminal Count = 1 when *n* clock cycles have been applied. The count can be reset by asserting the preset pin. This loads the counter with the appropriate value to produce a count sequence of length *n*.

Statistics

Device	Name	Speed (MHz)	Delay (ns)	Cells	Size (x * y)
AT40K	ctr16	85.0	11.8	6	1 x 6
AT40K	ctr8	98.9	10.1	5	1 x 5
AT94K/ AT40KAL	ctr16	85.0	11.8	6	1 x 6
AT94K/ AT40KAL	ctr8	95.0	10.5	5	1 x 5

Figure 5 shows an example of the ctr16 macro options.

Figure 5. Counter – Terminal Generator



IP Core Generator: Decoder

Features

- **Accessible from the Macro Generator Dialog and HDLPlanner™ – Included in IDS for FPGA Devices and System Designer™ for AT94K FPSLIC™ Devices**
- **Variable Output Logic Level**
- **Variable Width of Input Vectors**
- **Variable Width of Output Vectors**
- **Variable First Value to Decode**
- **Start Value Radix Selection**
- **Optional Register Inputs**
- **Optional Register Outputs**
- **Register Only**
  - **Variable Clock Inversion Capability**
  - **Initialization Polarity Selection**
  - **Registers Initialization Selection**

Description

The Decoder generator can be used to create a full or partial decode of the specified number of bits of input or output.

Programmable SLI
AT40K
AT40KAL
AT94K

Application Note

Rev. 2431B–1/02

Parameters

Parameter	Value	Explanation
Output	ActiveHigh	Output logic level will be held high for the decoded input
	ActiveLow	Output logic level will be held low for the decoded output
Input Width	Integer ≥ 0	Width of input vector. If this value is left at 0, the input width will be determined by the number of outputs specified.
Number of Values to Decode	Integer ≥ 0	Width of output vector. If this value is left at 0, the number of outputs will be determined by $2^{(1)}$ Input Width (i.e., all values will be decoded).
Starting at Value	Integer ≥ 0	First value to decode. When used in conjunction with the previous parameter, this can be used to specify exactly the range of outputs to decode. For example, setting the “Number of Values to Decode” parameter and Starting at Value” to 2 would decode the values 2, 3 and 4.
Radix of Start Value	Binary	“Starting at Value” parameter is specified as a binary number
	Octal	“Starting at Value” parameter is specified as an octal number
	Decimal	“Starting at Value” parameter is specified as a decimal number
	Hex	“Starting at Value” parameter is specified as a hexadecimal number
InRegister	Boolean	Register inputs on the decoder
OutRegister	Boolean	Register outputs on the decoder

Note: 1. At least one of Decodes or Width must be greater than 0 (unless Value is specified). If input or output registers are selected, three additional parameters are available.

Register Parameters

Parameter	Value	Explanation
Invert Clock	Boolean	Invert the register clock
Initialization Polarity = Low	Boolean	Make register initialization active low
Initialization	Reset	Registers can be reset to zero
	Set	Registers can be set to one
	None	Registers are initialized automatically

Pins

Type	Name	Option	Explanation
In	DATA[Width - 1:0]	No	Decoder input pins, either Width bits wide or log2 Decodes bits wide.
Out	EQ[Decodes - 1:0]	No	Decoder output, either 2 Width bits wide or decode bits wide. If both Width and Decodes are specified, the output will start at the lowest count and work up.
In	CLK/CLKN	Yes	Clock pin used for synchronous control (noninverted/inverted)
In	RN	Yes	Reset (active low)

Truth Table

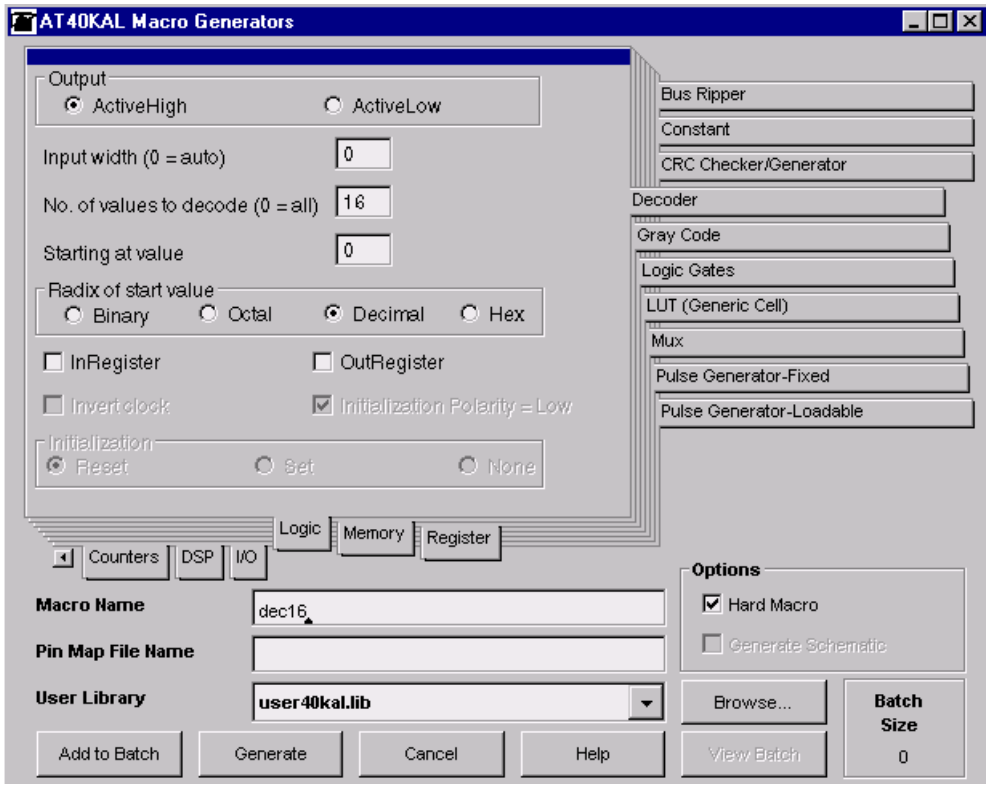
Option	Explanation
DATA[W - 1:0]	EQ[D - 1:0]
0	0...001
1	0...010
2	0...100
. . .	. . .

Statistics

Device	Name	Speed (MHz)	Delay (ns)	Cells	Size (x * y)
AT40K	dec16	339.0	3.0	16	1 x 16
AT40K	dec8	339.0	3.0	4	1 x 4
AT40KAL/ AT94KAL	dec16	492.6	2.0	16	1 x 16
AT40KAL/ AT94KAL	dec8	492.6	2.0	4	1 x 4

Figure 1 shows an example of the dec16 macro options.

Figure 1. Decoder Generator



IP Core Generator: Deductor

Features

- **Accessible from the Macro Generator Dialog and HDLPlanner™ – Included in IDS for FPGA Devices and System Designer™ for AT94K FPSLIC™ Devices**
- **Variable Pitch of Input Pins**
- **Variable Width of Output Vectors**
- **Registers Initialization Selection**
- **Variable Clock Inversion Capability**
- **Initialization Polarity Selection**
- **Preset Value Radix Selection**

Description

The Deductor subtracts a given number from the register initial value. The functional description of the deductor is as follows[1]:

```
always @ (posedge CLK or negedge R)
begin
  if(R == 'b0)
    SUM = 0;
  else if (ACC)
    {COUT, SUM} = SUM - DATA - CIN;
end
```

Note: 1. This code assumes that positive-edge clock and active-low reset have been specified.

Programmable SLI
AT40K
AT40KAL
AT94K

Application Note

Rev. 2432B–1/02

Parameters

Parameter	Value	Explanation
Pitch	Integer ≥ 1	Spacing between input pins, a pitch of 2 means will result in 1 cell between input pins.
Width	Integer > 1	Width of input and output vectors
Initialization	Reset	Registers can be reset to zero
	Set	Registers can be set to one
	None	Registers are set automatically on power-up
	Preset Value	Registers can be asynchronously loaded with a constant value
Invert Clock	Boolean	Inverts the clock input
Initialization Polarity = Low	Boolean	Set/Reset/Preset input is active low
Preset Value Radix	Binary	Constants for preset are specified in binary representation
	Octal	Constants for preset are specified in octal representation
	Decimal	Constants for preset are specified in decimal representation
	Hex	Constants for preset are specified in hexadecimal representation

Pins

Type	Name	Option	Explanation
In	CIN	No	Carry-in pin
In	ACCUMULATE	No	Enables the deductor, active high
In	DATA [Width - 1:0]	No	Data input
In	CLK/CLKN	Yes	Clock (noninverted/inverted)
In	R/RN/S/SN/P/PN	Yes	Reset/Set/Preset (active high/low)
Out	SUM[Width - 1:0]	No	Deductor output
Out	COUT	No	Carry-out pin(1)

Note: 1. Carry out = SUM[Width - 1:0] - DATA[Width - 1:0] - CIN < -2^n

Truth Table

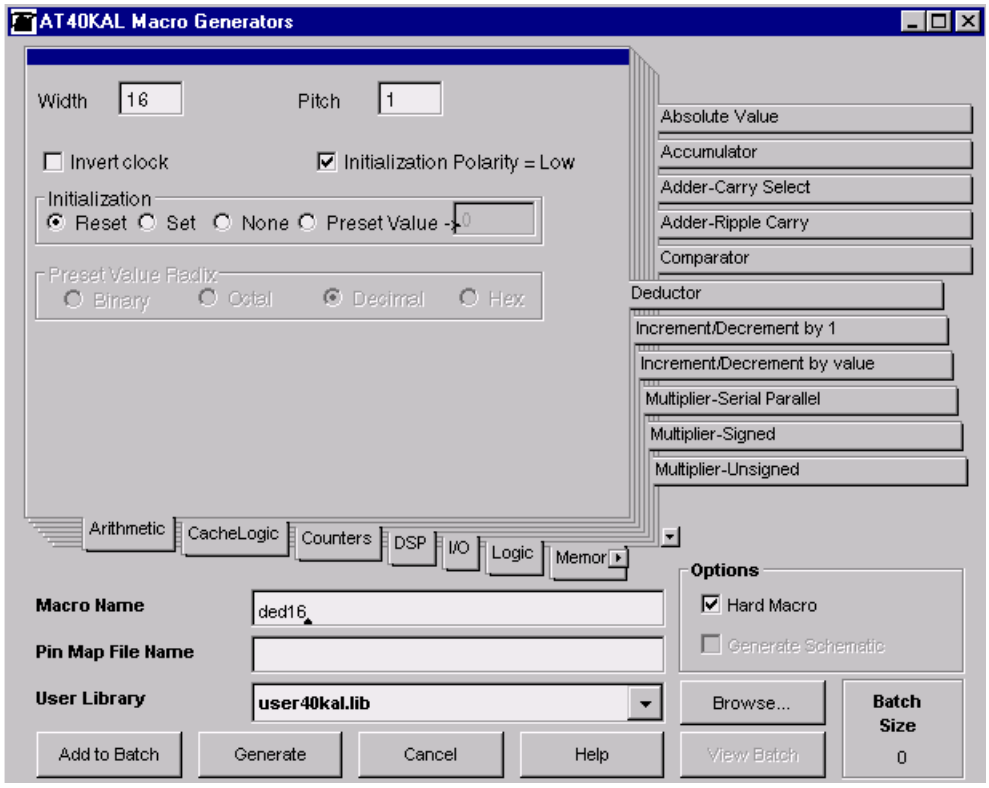
Input		Output	
CIN	DATA [W - 1:0]	SUM[W - 1:0]	COUT
A	B	SUM[W - 1:0] - A - B	1 if SUM[W - 1:0] - A - B < $-(2^W)$, 0 otherwise

Statistics

Device	Name	Speed (MHz)	Delay (ns)	Cells	Size (x * y)
AT40K	ded16	29.7	33.6	34	2 x 18
AT40K	ded8	48.9	20.4	18	2 x 10
AT40KAL/ AT94KAL	ded16	38.7	25.9	34	2 x 18
AT40KAL/ AT94KAL	ded8	61.5	16.3	18	2 x 10

Figure 1 shows an example of the ded16 macro options.

Figure 1. Deductor Generator



IP Core Generator: FIFO

Features

- **Accessible from the Macro Generator Dialog and HDLPlanner™ – Included in IDS for FPGA Devices and System Designer™ for AT94K FPSLIC™ Devices**
- **Variable Width of Parallel Input and Output Data**
- **Variable Depth of FIFO**
- **Variable Clock Inversion Capability**

Description

This generator creates a First-In First-Out (FIFO) buffer that makes use of the RAM cells in the AT40K Series architecture to produce a compact implementation with no read or write latency. In order to make efficient use of the RAM cells, some restrictions are placed on the data width and FIFO depth parameters: the data width must be a multiple of 4, and the FIFO depth is rounded up to the nearest power of 2.

The Read Enable (REN) and Write Enable (WEN) pins control the FIFO operation. When the WEN pin is asserted (i.e. pulled low), the buffer is in the write mode. Data presented at the D bus will be written into the FIFO until the FULL pin goes low, indicating that the FIFO cannot accept any more data. When the REN pin is asserted, the FIFO is in the read mode. Data can be read from the Q bus until the EMPTY pin goes low, indicating that no more data is present in the buffer. When REN and WEN are both high, operation of the FIFO is effectively suspended.

Parameters

Parameter	Value	Explanation
Data Width	Integer ≥ 4	Width of parallel input and output data (must be a multiple of 4)
Depth	Integer > 2	FIFO depth (rounded up to the nearest power of 2)
Invert Clock	Boolean	Invert the polarity of the clock input

Programmable SLI
AT40K
AT40KAL
AT94K

Application Note

Rev. 2433B–1/02

Pins

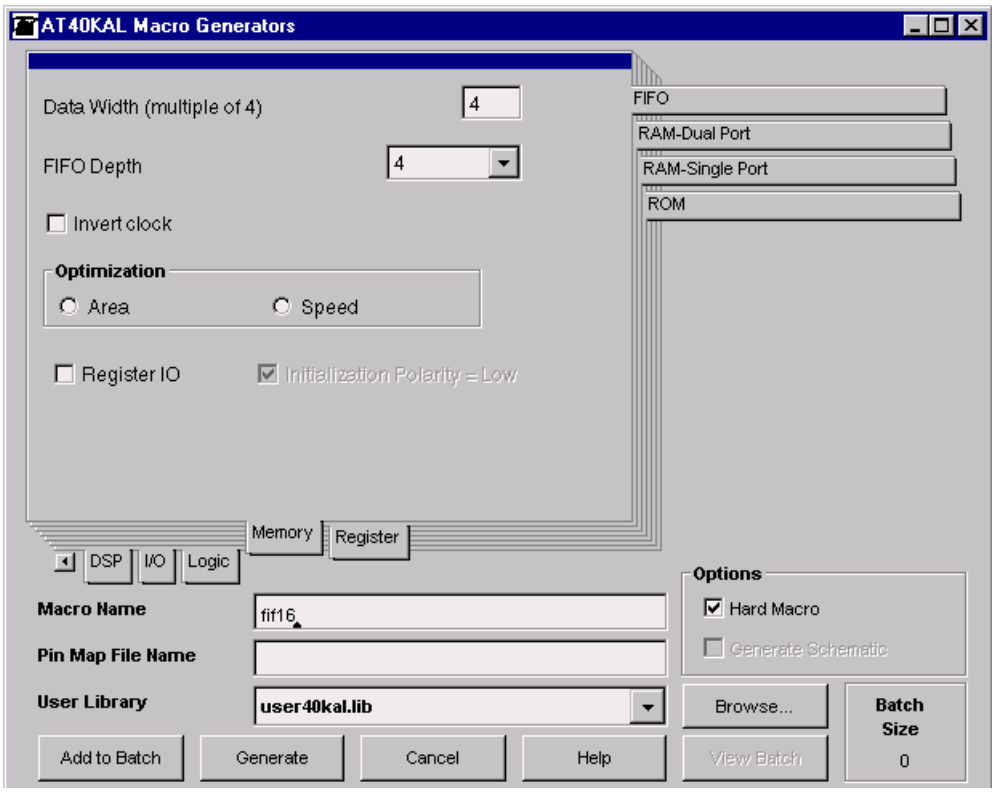
Type	Name	Option	Explanation
In	D [Width - 1:0]	No	Data input bus
Out	Q [Width - 1:0]	No	Data output bus
In	CLK/CLKN	No	Clock (noninverted/inverted)
In	REN	No	Read enable pin (active low)
In	WEN	No	Write enable pin (active low)
In	RESET	Yes	Reset (active low)
Out	FULL	No	FIFO full flag (active low)
Out	EMPTY	No	FIFO empty flag (active low)
Out	SH	Yes	Indicator of bang band register output
Out	MQ	Yes	Bang band register output
Out	SQ	Yes	Shift register output

Statistics

Device	Name	Speed (MHz)	Delay (ns)	Cells	Size (x * y)
AT40K	fif16	38.3	26.3	19	7 x 6
AT40K	fif8	38.0	26.1	16	7 x 5
AT40KAL/ AT94KAL	fif16	46.0	21.7	19	7 x 6
AT40KAL/ AT94KAL	fif8	45.8	21.8	16	7 x 5

Figure 1 shows an example of the fif16 macro options.

Figure 1. FIFO Generator



IP Core Generator: Flip-Flop

Features

- **Flip-Flop – D-Type**
- **Flip-Flop – Toggle**
- **Accessible from the Macro Generator Dialog and HDLPlanner™ – Included in IDS for FPGA Devices and System Designer™ for AT94K FPSLIC™ Devices**
- **Register Bank Enable Selection**
- **Tri-state Control Selection**
- **Variable Width of Input and Output Data**
- **Variable Pitch of Input Pins**
- **Variable Clock Inversion Capability**
- **Initialization Polarity Selection**
- **Asynchronous Initialization Selection**
- **Synchronous Initialization Selection**
- **Initialization Value Radix Selection**

Flip-Flop – D-Type

The D Flip-Flop generator can be used to create a register bank consisting of D-type flip-flops.

Programmable SLI
AT40K
AT40KAL
AT94K

Application Note

Rev. 2434B–1/02

Parameters

Parameter	Value	Explanation
Register Bank Enable	None	No register enable control is provided
	Group Enable	A single enable input is used to control all flip-flops
	Individual Enables	Each flip-flop in the register bank has its own enable control
Tri-state Control	None	Register bank has no tri-state control
	Group OE pin	A single OE pin is used to tri-state the outputs of all flip-flops
	Individual OE pins	Each flip-flop has its own tri-state control
Width	Integer > 0	Width of the input and output data
Pitch	Integer > 0	Spacing between input pins. Pitch of 2 means one cell between inputs
Invert Clock	Boolean	Inverts the clock input
Initialization Polarity = Low	Boolean	Reset/Set/Preset input is active low
Asynchronous Initialization	Reset	Registers can be reset to zero
	Set	Registers can be set to one
	None	Registers are automatically reset on power-up
	Preset	Registers can be asynchronously loaded with a constant value
Synchronous Initialization	None	Registers have synchronous initialization capability
	Auxiliary Input	Registers have a second data input that can be used to synchronously preset them to any value
	Constant Value	Registers can be synchronously preset with a user-supplied constant value
Initialization Value Radix	Binary	Initialization values are specified using binary representation
	Octal	Values are specified in octal
	Decimal	Values are specified in decimal
	Hex	Values are specified in hexadecimal

Pins

Type	Name	Option	Explanation
In	DATA[Width - 1:0]	No	Data input
In	CLK/CLKN	No	Clock pin (noninverted/inverted)
Out	Q[Width - 1:0]	No	Data output
In	ENABLE	Yes	Group enable input
In	ENABLE[Width - 1:0]	Yes	Individual enable inputs
In	LOAD	Yes	Perform synchronous initialization
In	LOADDATA[Width - 1: 0]	Yes	Data input for synchronous initialization
In	OE	Yes	Group tri-state control input
In	OE[Width - 1:0]	Yes	Individual tri-state control inputs
In	R/RN/S/SN/P/PN	No	Reset/Set/Preset (active high/low)

Truth Table(1)

Input				Output(2)
RN	DATA[W - 1:0]	CLK	ENABLE	Q[W - 1:0]
0	X	X	X	0
1	X	0>1	X	No Change
1	0	0>1	1	0
1	1	0>1	1	1
1	X	0>1	1	Q(I) = Q - (I - 1) Q(0) = SI

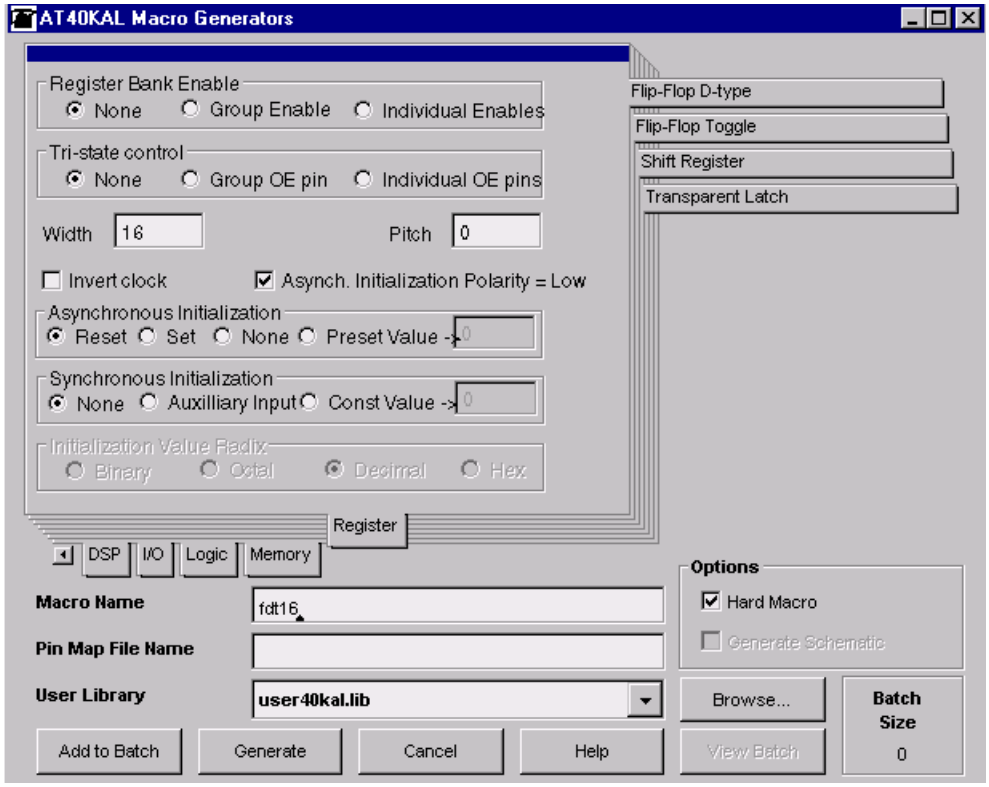
Notes: 1. This Truth Table assumes that an active-low reset and noninverted clock have been selected.
2. Q - is the value of Q preceding the clock transition.

Statistics

Device	Name	Speed (MHz)	Delay (ns)	Cells	Size (x * y)
AT40K	fdt16	565.0	1.8	16	1 x 16
AT40K	fdt8	565.0	1.8	8	1 x 8
AT40KAL/ AT94KAL	fdt16	598.8	1.7	16	1 x 16
AT40KAL/ AT94KAL	fdt8	598.8	1.7	8	1 x 8

Figure 1 shows an example of the fdt16 macro options.

Figure 1. Flip-Flop – D-Type Generator



Flip-Flop – Toggle

The Toggle Flip-Flop generator can be used to create a register bank consisting of T-type (toggle) flip-flops.

Parameters

Parameter	Value	Explanation
Register Bank Enable	None	No register enable control is provided
	Group Enable	A single enable input is used to control all flip-flops
	Individual Enables	Each flip-flop in the register bank has its own enable control
Tri-state Control	None	Register bank has no tri-state control
	Group OE pin	A single OE pin is used to tri-state the outputs of all flip-flops
	Individual OE pins	Each flip-flop has its own tri-state control
Width	Integer > 0	Width of the input and output data
Pitch	Integer > 0	Spacing between input pins. Pitch of 2 means one cell between inputs.
Invert Clock	Boolean	Inverts the clock input
Initialization Polarity = Low	Boolean	Reset/Set/Preset input is active low
Asynchronous Initialization	Reset	Registers can be reset to zero
	Set	Registers can be set to one
	None	Registers are automatically reset on power-up
	Preset	Registers can be asynchronously loaded with a constant value
Synchronous Initialization	None	Registers have synchronous initialization capability
	Auxiliary Input	Registers have a second data input that can be used to synchronously preset them to any value
	Constant Value	Registers can be synchronously preset with a user-supplied constant value
Initialization Value Radix	Binary	Initialization values are specified using binary representation
	Octal	Values are specified in octal
	Decimal	Values are specified in decimal
	Hex	Values are specified in hexadecimal

Pins

Type	Name	Option	Explanation
In	DATA[Width - 1:0]	No	Data input
In	CLK/CLKN	No	Clock pin (noninverted/inverted)
Out	Q[Width - 1:0]	No	Data output
In	ENABLE	Yes	Group enable input
In	ENABLE[Width - 1:0]	Yes	Individual enable inputs
In	LOAD	Yes	Performs synchronous initialization
In	LOADDATA[Width - 1: 0]	Yes	Data input for synchronous initialization
In	OE	Yes	Group tri-state control input
In	OE[Width - 1:0]	Yes	Individual tri-state control inputs
In	R/RN/S/SN/P/PN	No	Reset/Set/Preset (active high/low)

Truth Table(1)

Input				Output(2)
RN	DATA[W - 1:0]	CLK	ENABLE	Q[W - 1:0]
0	X	X	X	0
1	X	0>1	0	No Change
1	0	0>1	1	Q
1	1	0>1	1	~Q

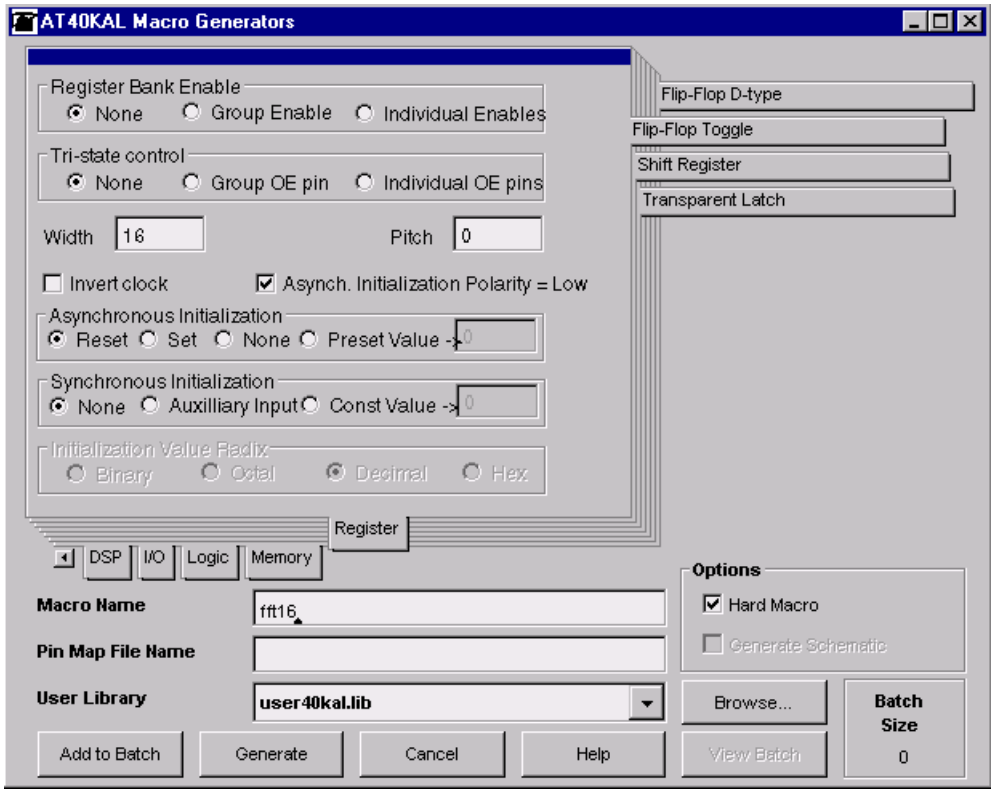
Notes: 1. This Truth Table assumes that an active-low reset and non inverted clock have been selected.
2. Q - is the value of Q preceding the clock transition.

Statistics

Device	Name	Speed (MHz)	Delay (ns)	Cells	Size (x * y)
AT40K	fft16	565.0	1.8	16	1 x 16
AT40K	fft8	565.0	1.8	8	1 x 8
AT40KAL/ AT94KAL	fft16	598.8	1.7	16	1 x 16
AT40KAL/ AT94KAL	fft8	598.8	1.7	8	1 x 8

Figure 2 shows an example of the fft16 macro options.

Figure 2. Flip-Flop – Toggle Generator



IP Core Generator: Gray Code

Features

- **Accessible from the Macro Generator Dialog and HDLPlanner™ – Included in IDS for FPGA Devices and System Designer™ for AT94K FPSLIC™ Devices**
- **Function Code Selection**
- **Variable Width of Input and Output Data**

Description

The Gray Code generator can be used to convert binary code to gray code or gray code to binary code respectively.

Parameters

Parameter	Value	Explanation
Function	ToGray	Converts the binary code input to gray code
	ToBinary	Converts the gray code input to binary code
Width	Integer > 1	Width of input and output data

Pins

Type	Name	Option	Explanation
In	DATA[Width - 1:0]	No	Input data to be converted
Out	OUT[Width - 1:0	No	Output bits

Truth Table

Input		Output
Binary	DATA[W - 1:0]	DATAW, DATA[W - 1] xor DATAW, DATA[W - 2] xor DATA[W - 1],...
Gray	DATA[W - 1:0]	DATAW, DATA[W - 1] xor DATAW, DATA[W - 2] xor DATA[W - 1] xor DATAW, ...

Programmable SLI
AT40K
AT40KAL
AT94K

Application Note

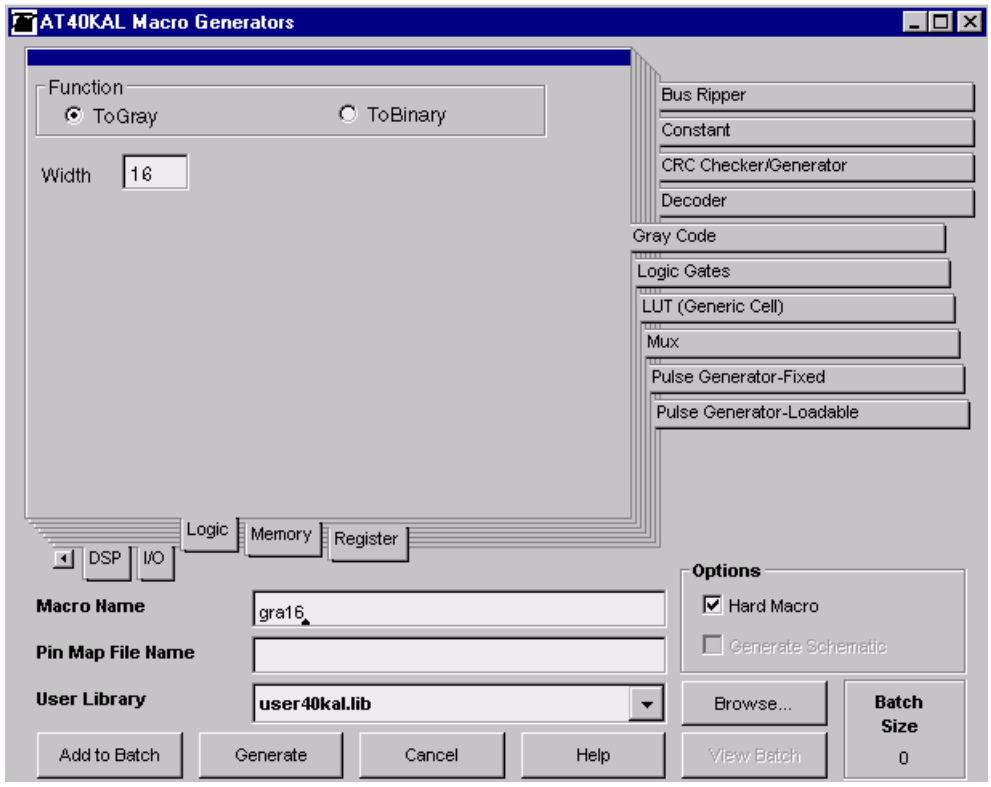
Rev. 2435B–1/02

Statistics

Device	Name	Speed (MHz)	Delay (ns)	Cells	Size (x * y)
AT40K	gra16	423.7	2.4	8	1 x 8
AT40K	gra8	423.7	2.4	4	1 x4
AT40KAL/ AT94KAL	gra16	543.5	1.8	8	1 x 8
AT40KAL/ AT94KAL	gra8	543.5	1.8	4	1 x4

Figure 1 shows an example of the gra16 macro options.

Figure 1. Gray Code Generator



IP Core Generator: Incrementor/Decrementor

Features

- **Incrementor/Decrementor by 1**
- **Incrementor/Decrementor by Value**
- **Accessible from the Macro Generator Dialog and HDLPlanner™ – Included in IDS for FPGA Devices and System Designer™ for AT94K FPSLIC™ Devices**
- **Incrementor/Decrementor Generation Selection**
- **Variable Pitch of Input Pins**
- **Variable Width of Input and Output Vectors**
- **Parallel Load Capability**
- **Optional Enable**
- **Initialization Polarity Selection**
- **Clock Inversion Capability**
- **Initialization Polarity Capability**
- **Preset Value Radix Selection**

Incrementor/Decrementor by 1

The Incrementor/Decrementor by 1 generator can be used to add/subtract 1 to/from the current registered value. This generator can be used to create a macro which increments or decrements by 1 a preloaded value or the data input on each rising (or falling) edge of the clock. The functional description of the generator is as follows[1]:

```
always @(posedge CLK or negedge R)
    begin
  if(R == 'b0)
      SUM = 0;
  else if (ENABLE && !LOAD) // Assuming Load is active low
      SUM = DATA;
  else if (ENABLE)
      begin
          if (INC_DEC)
      {COUT, SUM} = SUM + 1;
          else
      {COUT, SUM} = SUM - 1;
      end
    end
```

Note: 1. This code assumes that positive-edge clock and active-low reset have been specified.

Programmable SLI
AT40K
AT40KAL
AT94K

Application Note

Rev. 2436B–1/02

Parameters

Parameter	Value	Explanation
Function	Increment	Generates an incrementor
	Decrement	Generates a decrementor
	Increment Decrement	Generates a programmable incrementor/decrementor
Pitch	Integer ≥ 1	Spacing between input pins. A pitch of 2 means one cell between input pins.
Width	Integer > 1	Width of input and output vectors
Load	Boolean	Macro has a parallel load capability
Enable	Boolean	Provides enable pin
Initialization	Reset	Registers can be reset to zero
	Set	Registers can be set to one
	None	Registers are automatically reset on power-up
	Preset	Registers can be asynchronously loaded with a constant value
Invert Clock	Boolean	Inverts the clock input
Initialization Polarity = Low	Boolean	Initialization input is active low
Preset Value Radix	Binary	Constants for preset are specified in binary representation
	Octal	Constants are specified in octal
	Decimal	Constants are specified in decimal
	Hex	Constants are specified in hexadecimal

Pins

Type	Name	Option	Explanation
In	LOADIN	Yes	Loads the data if low
In	ENABIN	Yes	Enables the incrementor/decrementor, active high
In	INC_DEC	Yes	Incrementor if high, otherwise decrementor
In	DATA[Width - 1:0]	No	Data input, if LOADIN is low, as parallel load
In	CLK/CLKN	Yes	Clock (noninverted/inverted)
In	R/RN/S/SN/P/PN	Yes	Reset/Set/Preset (active high/low)
Out	SUM[Width - 1:0]	No	Accumulator output
Out	COUT	No	Carry out(1)

Note: 1. If Incrementor, then Carry out = SUM[Width - 1:0] + 1 > 2^n - 1
else Carry out = SUM[Width - 1:0] - 1 < -2^n

Truth Table

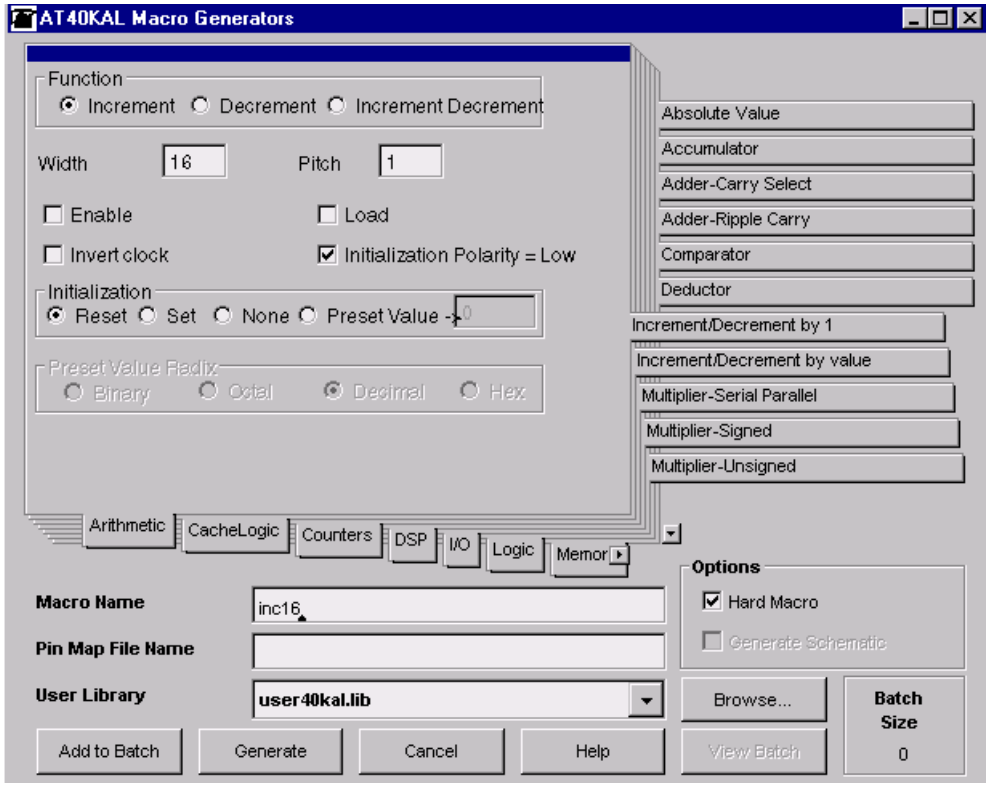
Input				Output	
LOADIN	ENABIN	INC_DEC	DATA [W - 1:0]	SUM[W - 1:0]	COUT
0	x	1	A	A	1 if A + 1> (2^W) - 1, 0 otherwise
0	x	0	A	A	1 if A - 1< -(2^W), 0 otherwise
1	0	x	x	Present State	Present State
1	1	1	x	SUM[W - 1:0] + 1	1 if SUM[W - 1:0] + 1 > (2^W) - 1, 0 otherwise
1	1	0	x	SUM[W - 1:0] - 1	1 if SUM[W - 1:0] - 1 > (2^W) - 1, 0 otherwise

Statistics

Device	Name	Speed (MHz)	Delay (ns)	Cells	Size (x * y)
AT40K	inc16	33.7	29.7	17	1 x 17
AT40K	inc8	60.8	16.5	9	1 x 9
AT40KAL/ AT94KAL	inc16	39.4	25.4	17	1 x 17
AT40KAL/ AT94KAL	inc8	64.8	15.4	9	1 x 9

Figure 1 shows an example of the inc16 macro options.

Figure 1. Incrementor/Decrementor by 1 Generator



Incrementor/ Decrementor by Value

The Incrementor/Decrementor by value generator can be used to add/subtract a value to/from the current registered value. This generator can be used to create a macro which increments or decrements a preloaded value or the data input by a user-specified amount on each rising (or falling) edge of the clock. The functional description of the generator is as follows(1):

```
always @(posedge CLK or negedge R)
    begin
  if(R == 'b0)
      SUM = 0;
  else if (ENABLE && !LOAD)// Assuming Load is active low
      SUM = DATA;
  else if (ENABLE)
      begin
          if (INC_DEC)
      {COUT, SUM} = SUM + VALUE;
          else
      {COUT, SUM} = SUM - VALUE;
      end
    end
```

Note: 1. This code assumes that positive-edge clock and active-low reset have been specified.

Parameters

Parameter	Value	Explanation
Function	Increment	Generates an incrementor
	Decrement	Generates a decrementor
	Increment Decrement	Generates a programmable incrementor/decrementor
Pitch	Integer ≥ 1	Spacing between input pins. A pitch of 2 means one cell between input pins.
Width	Integer > 1	Width of input and output vectors
Incrementor/ Decrementor Value	Integer ≥ 1	Amount to increment/decrement by
Load	Boolean	Macro has a parallel load capability
Enable	Boolean	Provides enable pin
Initialization	Reset	Registers can be reset to zero
	Set	Registers can be set to one
	None	Registers are automatically reset on power-up
	Preset	Registers can be asynchronously loaded with a constant value
Invert Clock	Boolean	Inverts the clock input
Initialization Polarity = Low	Boolean	Initialization input is active low

Parameters (Continued)

Parameter	Value	Explanation
Preset Value Radix	Binary	Constants for preset are specified in binary representation
	Octal	Constants are specified in octal
	Decimal	Constants are specified in decimal
	Hex	Constants are specified in hexadecimal

Pins

Type	Name	Option	Explanation
In	LOADIN	Yes	Loads the data if low
In	ENABIN	Yes	Enables the incrementor/decrementor, active high
In	INC_DEC	Yes	Incrementor if high, otherwise decrementor
In	DATA[Width - 1:0]	No	Data input, if LOADIN is low, as parallel load
In	CLK/CLKN	Yes	Clock (noninverted/inverted)
In	R/RN/S/SN/P/PN	Yes	Reset/Set/Preset (active high/low)
Out	SUM[Width - 1:0]	No	Accumulator output
Out	COUT	No	Carry out(1)

Note: 1. If Incrementor, then Carry out = SUM[Width - 1:0] + IncDecValue > 2^n - 1
else Carry out = SUM[Width - 1:0] - IncDecValue < -2^n

Truth Table

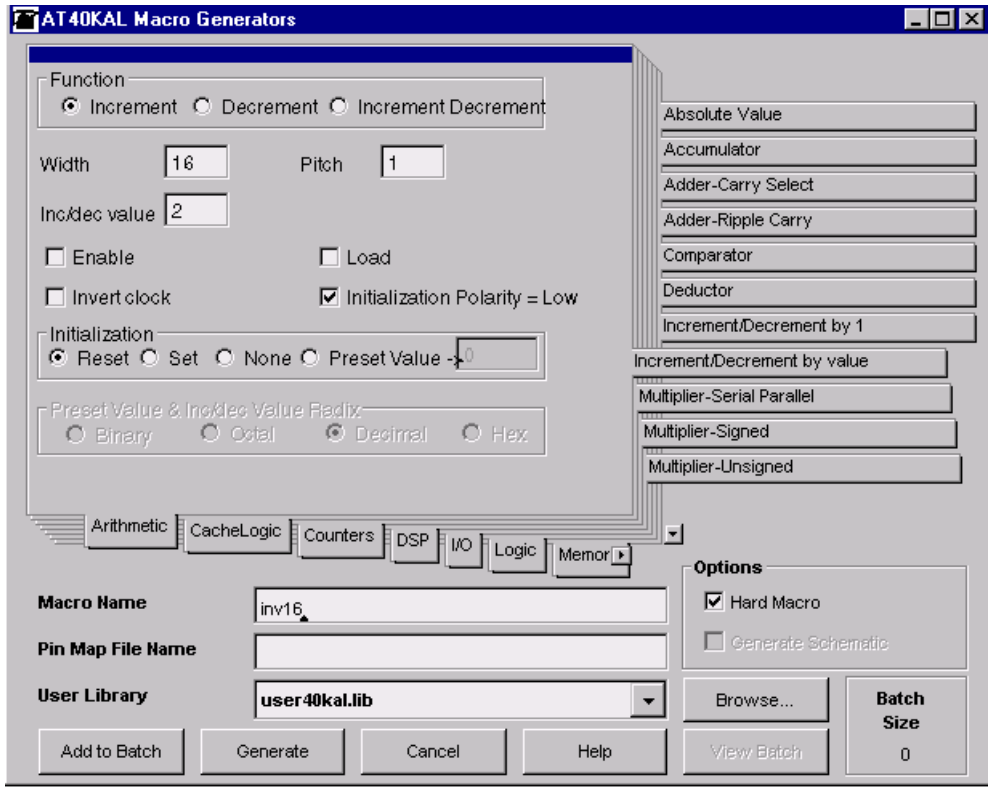
Input				Output	
LOADIN	ENABIN	INC_DEC	DATA [W - 1:0]	SUM[W - 1:0]	COUT
0	x	1	A	A	1 if A + IncDecValue > (2^W) - 1, 0 otherwise
0	x	0	A	A	1 if A - IncDecValue < $-(2^W)$, 0 otherwise
1	0	x	x	Present State	Present State
1	1	1	x	SUM[W - 1:0] + IncDecValue	1 if SUM[W - 1:0] + IncDecValue > (2^W) - 1, 0 otherwise
1	1	0	x	SUM[W - 1:0] - IncDecValue	1 if SUM[W - 1:0] - IncDecValue < $-(2^W)$, 0 otherwise

Statistics

Device	Name	Speed (MHz)	Delay (ns)	Cells	Size (x * y)
AT40K	inv16	31.9	31.3	17	1 x 17
AT40K	inv8	55.2	18.1	9	1 x 9
AT40KAL/ AT94KAL	inv16	37.6	26.6	17	1 x 17
AT40KAL/ AT94KAL	inv8	60.0	16.7	9	1 x 9

Figure 2 shows an example of the inv16 macro options.

Figure 2. Incrementor/Decrementor by Value Generator



IP Core Generator: Transparent Latch

Features

- **Accessible from the Macro Generator Dialog and HDLPlanner™ – Included in IDS for FPGA Devices and System Designer™ for AT94K FPSLIC™ Devices**
- **Variable Width of Input and Output Data**
- **Variable Pitch of Input Pins**
- **Latch Enable Selection**
- **Tri-state Control Selection**
- **Reset/Set/Preset Pin Selection**
- **Asynchronous Initialization Selection**
- **Initialization Value Radix Selection**

Description

The Latch generator can be used to create a D-type transparent latch.

Programmable SLI
AT40K
AT40KAL
AT94K

Application Note

Rev. 2437B–1/02

Parameters

Parameter	Type	Explanation
Width	Integer > 0	Width of the input and output data
Pitch	Integer > 0	Spacing between input pins. Pitch of 2 means one cell between inputs
Latch Enable	Group Enable	One enable pin controls all latches
	Individual Enables	Each latch has its own enable input
Tri-state Control	None	Latch outputs are not tri-stated
	Group OE pin	Latch outputs are tri-stated and controlled from a single OE pin
	Individual OE pins	Each latch has its own OE pin to control the tri-stating of its outputs
Active Low Set/Reset	Boolean	Reset/Set/Preset input is active low
Asynchronous Initialization	Reset	Latches can be reset to zero
	Set	Latches can be set to one
	None	Latches are automatically reset on power-up
	Preset	Latches can be asynchronously loaded with a constant value
Initialization Value Radix	Binary	Initialization values are specified using binary representation
	Octal	Values are specified in octal
	Decimal	Values are specified in decimal
	Hex	Values are specified in hexadecimal

Pins

Type	Name	Option	Explanation
In	DATA[Width - 1:0]	No	Data input to the D-type latches
In	ENABLE	Yes	Common Latch Enable input (1 = flow-through, 0 = latch)
In	ENABLE[Width - 1:0]	Yes	Individual Latch Enable inputs (1 = flow-through, 0 = latch)
In	OE	Yes	Common Output Enable input (0 = tri-state)
In	OE[Width - 1:0]	Yes	Individual Output Enable inputs (0 = tri-state)
In	R/RN/S/SN/P/PN	Yes	Reset/Set/Preset (active high/low)
Out	Q[Width - 1:0]	No	Data output from D-type latches

Statistics

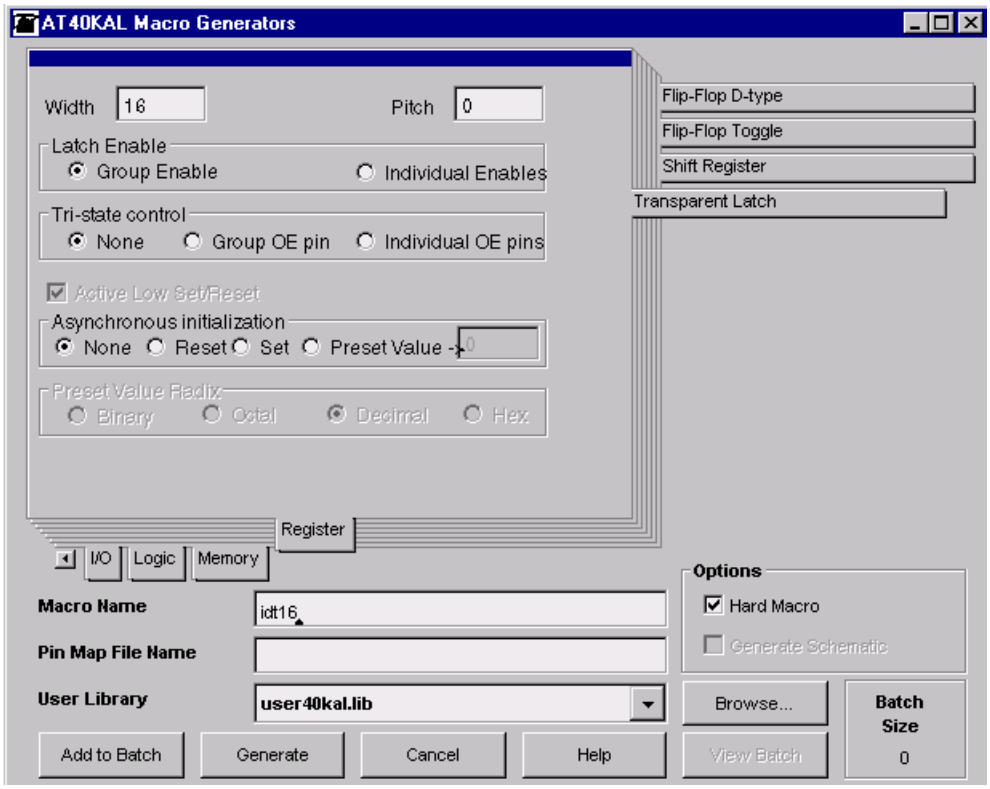
Name	Speed (MHz)	Delay (ns)	Cells	Size (x * y)
Idt16	476.2	2.1	16	1 x 16
Idt8	476.2	2.1	8	1 x 8

Statistics

Device	Name	Speed (MHz)	Delay (ns)	Cells	Size (x * y)
AT40K	Idt16	476.2	2.1	16	1 x 16
AT40K	Idt8	476.2	2.1	8	1 x 8
AT40KAL/ AT94KAL	Idt16	719.4	1.4	16	1 x 16
AT40KAL/ AT94KAL	Idt8	719.4	1.4	8	1 x 8

Figure 1 shows an example of the idt16 macro options.

Figure 1. Latch – Transparent Generator



IP Core Generator: Logic Gates

Features

- **Accessible from the Macro Generator Dialog and HDLPlanner™ – Included in IDS for FPGA Devices and System Designer™ for AT94K FPSLIC™ Devices**
- **Variable Width of Input and Output Data**
- **Variable Pitch of Input Pins**
- **Latch Enable Selection**
- **Tri-state Control Selection**
- **Reset/Set/Preset Pin Selection**
- **Asynchronous Initialization Selection**
- **Initialization Value Radix Selection**

Description

A variety of Logic Gates can be generated, each with a programmable number of inverted inputs.

Parameters

Parameter	Value	Explanation
Gate Type	AND	AND gates
	NAND	NAND gates
	OR	OR gates
	NOR	NOR gates
	XOR	Exclusive OR gates
	XNOR	Exclusive NOR gates
	INV	Inverter gates
Number of Gates	Integer > 0	Number of gates in component
Number of Inputs per Gate	Integer > 0	Width of input vector for each gate
Number of Inverted Inputs per Gate	Integer > 0	Number of inputs that are inverted on each gate. The inverted inputs start from the first input to the gate and count up.
Common Input	Boolean	Select this option if a common input should be supplied to each of the gates

Programmable SLI
AT40K
AT40KAL
AT94K

Application Note

Rev. 2438A–1/02

Pins

Type	Name	Option	Explanation
In	DATA[S - 1:0]_[W - 1:0]	No	Gate input pin where S(1) goes from 0 to Size and W(2) goes from 0 to Width
Out	RESULT[S - 1:0]	No	Gate output
In	COMMON_DATA	Yes	Common input Enable pin for each of the gates

Notes: 1. S stands for Size (number of gates).
2. W stands for Width (number of inputs per gate).

Truth Table

Type	Name(1)	Explanation(2)
AND	DATA[S - 1:0]_[W - 1:0]	DATA0 * DATA1 ... * DATAW - 1
INV	DATA[S - 1:0]_[W - 1:0]	~DATA0, ~DATA1, ... ~DATAS - 1
NAND	DATA[S - 1:0]_[W - 1:0]	~(DATA0 * DATA1 ... * DATAW - 1)
NOR	DATA[S - 1:0]_[W - 1:0]	~(DATA0 + DATA1 ... + DATAW - 1)
OR	DATA[S - 1:0]_[W - 1:0]	DATA0 + DATA1 ... + DATAW - 1
XOR	DATA[S - 1:0]_[W - 1:0]	DATA0 ^ DATA1 ... ^ DATAW - 1
XNOR	DATA[S - 1:0]_[W - 1:0]	~(DATA0 ^ DATA1 ... ^ DATAW - 1)

Notes: 1. S stands for Size (number of gates).
2. W stands for Width (number of inputs per gate).

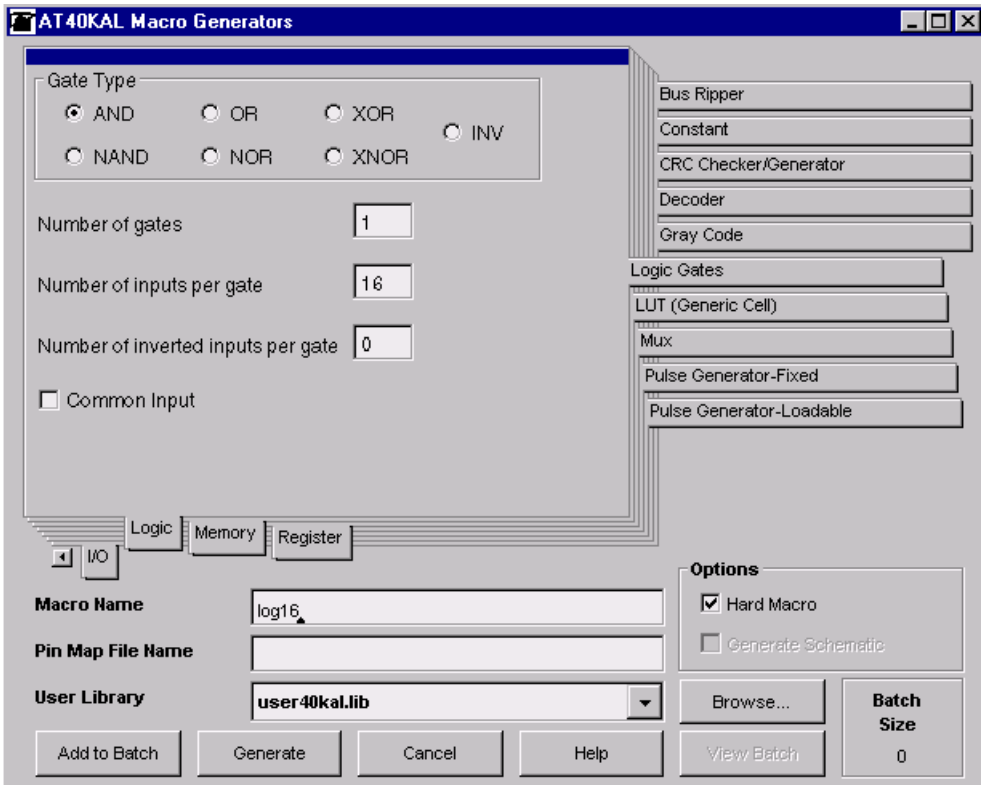
If the inverter function is selected, or the gate width is 2, the macro must be generated as a soft macro (i.e., with the Hard macro option deselected). This is to ensure the most efficient implementation of the logic as small gates are better handled by the Mapper in Figaro and cannot be effectively implemented as a hard macro.

Statistics

Device	Name	Speed (MHz)	Delay (ns)	Cells	Size (x * y)
AT40K	log16	96.9	10.4	6	4 x 2
AT40K	log8	189.9	5.3	3	3 x 2
AT40KAL/ AT94KAL	log16	138.9	7.2	6	4 x 2
AT40KAL/ AT94KAL	log8	304.9	3.3	3	3 x 2

Figure 1shows an example of the log16 macro options.

Figure 1. Logic Gates Generators



IP Core Generator: Look-Up Table

Features

- **Accessible from the Macro Generator Dialog and HDLPlanner™ – Included in IDS for FPGA Devices and System Designer™ for AT94K FPSLIC™ Devices**
- **Look-Up Table Selection**
- **Tri-state Control Selection**
- **Variable Number of Look-Up Tables**
- **Variable Pitch of Generic Cells**
- **Internal Feedback Capability**
- **Function G Capability**
- **Function H Capability**
- **Optional Register**
- **Clock Inversion Capability**
- **Initialization Polarity Selection**
- **Initialization Selection**
- **Initialization Value Radix Selection**

Description

The Look-Up Table (LUT), or Generic Cell generator can be used as a convenient interface for generating components using the FGENxx and MGENxx generic cell components in the AT40K macro library. The generator allows complete control over the function of an AT40K core cell, and it allows this function to be replicated *n* times.

Programmable SLI
AT40K
AT40KAL
AT94K

Application Note

Rev. 2439A–1/02

Parameters

Parameter	Value	Explanation
LUT Type	4-input LUT	Configure each cell in the macro as an FGEN1-type component, with a single 4-input LUT.
	2 x 3-input LUT	Configure each cell in the macro as an FGEN2-type component, with two 3-input LUT sharing common inputs.
	4-input LUT with AND gate	Configure each cell in the macro as an MGEN-type component, with a single 4-input LUT and an upstream AND gate.
Tri-state Control	None	Cells have no tri-state control
	Group OE pin	A single OE pin is used to tri-state the outputs of all cells
	Individual OE pins	Each cell has its own tri-state control
Number of LUTs	Integer > 0	Number of generic cells to be included in the macro, each cell is configured identically
Pitch	Integer > 0	Spacing between generic cells
Internal Feedback	Boolean	Program the cell to include internal feedback (FB input). This reduces the number of external inputs to the cell by one. Not available for 1 x 4-input LUT with AND gate.
Function G	String	Equation string used to specify the function of the G cell output, in terms of A, B, C, D and FB (if feedback option is selected).
Function H	String	Equation string used to specify the function of the H cell output, in terms of A, B, C, D and FB (if feedback option is selected).
Register	Boolean	Include a register in the cell

If the register option is selected, the following additional parameters are available:

Register Parameters

Parameter	Value	Explanation
Invert Clock	Boolean	Invert the clock input
Initialization Polarity = Low	Boolean	Reset/Set/Preset input is active low
Initialization	Reset	Registers can be reset to zero
	Set	Registers can be set to one
	None	Registers are automatically reset on power-up
	Preset	Registers can be asynchronously loaded with a constant value

Register Parameters (Continued)

Parameter	Value	Explanation
Initialization Value Radix	Binary	Initialization value is specified using binary representation
	Octal	Value is specified in octal
	Decimal	Value is specified in decimal
	Hex	Value is specified in hexadecimal

Pins

Type	Name[1]	Option	Explanation
In	A[N - 1:0]	No	Data input A (must always be used in equation)
In	B[N - 1:0]	Yes	Data input B
In	C[N - 1:0]	Yes	Data input C
In	D[N - 1:0]	Yes	Data input D
Out	G[N - 1:0]	No	Data output
Out	H[N - 1:0]	Yes	Data output
In	OE	Yes	Group output enable pin
In	OE[N - 1:0]	Yes	Individual output enable pins
In	CLK/CLKN	No	Clock pin (noninverted/inverted)
In	R/RN/S/SN/P/PN	No	Reset/Set/Preset (active high/low)

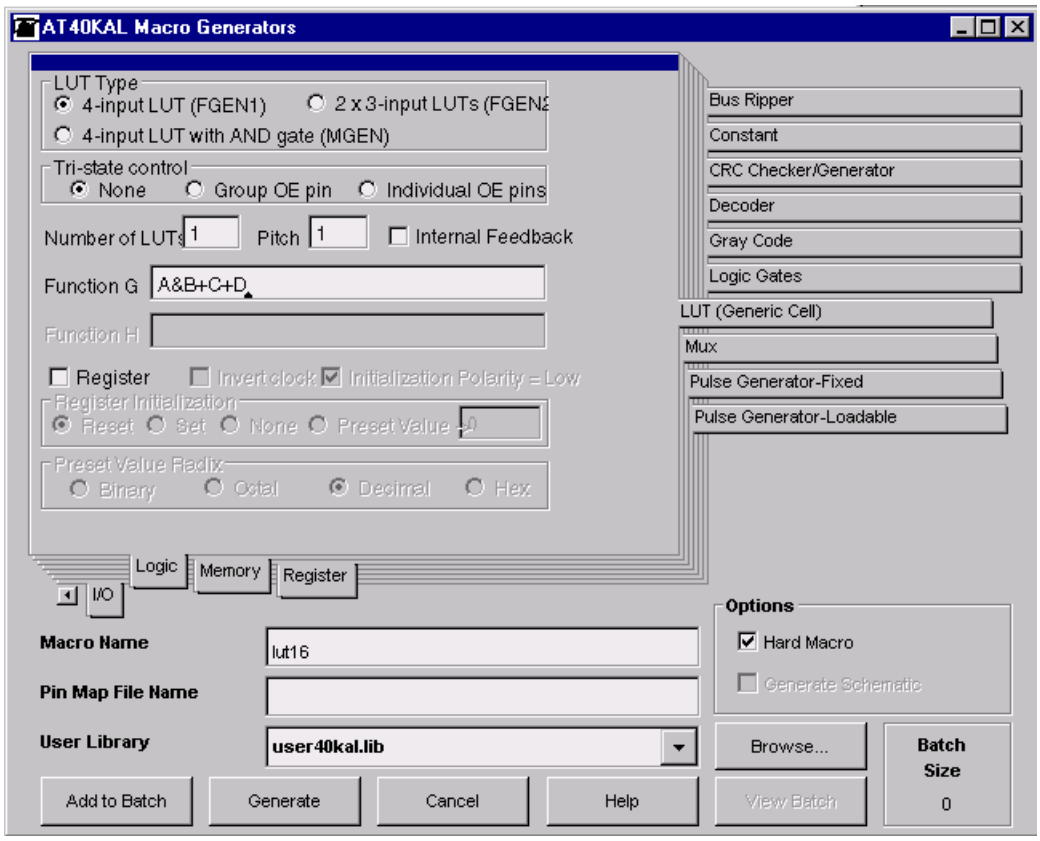
Note: 1. N stands for the number of generic cells in the macro.

Statistics

Device	Name	Speed (MHz)	Delay (ns)	Cells	Size (x * y)
AT40K	lut16	339.0	3.0	16	1 x 16
AT40K	lut8	339.0	3.0	8	1 x 8
AT40KAL/ AT94KAL	lut16	492.6	2.0	16	1 x 16
AT40KAL/ AT94KAL	lut8	492.6	2.0	8	1 x 8

Figure 1 shows an example of the lut16 macro options.

Figure 1. Look-Up Table Generator



IP Core Generator: Multiplier

Features

- Multiplier – Serial Parallel
- Multiplier – Signed
- Multiplier – Signed, Pipeline x 1
- Multiplier – Unsigned
- Multiplier – Unsigned, Pipeline x 1
- Accessible from the Macro Generator Dialog and HDLPlanner™ – Included in IDS for FPGA Devices and System Designer™ for AT94K FPSLIC™ Devices
- Serial Parallel Multiplier Only
  - Variable Width of Input Data
  - Input Representation Selection
  - Clock Inversion Capability
  - Initialization Polarity Selection
- Signed Multiplier Only
  - Variable Width of Input DataA
  - Variable Width of Input DataB
- Signed Multiplier, Pipeline x 1 Only
  - Variable Width of Input DataA
  - Variable Width of Input DataB
  - Initialization Polarity Selection
  - Initialization Selection
- Unsigned Multiplier Only
  - Variable Width of Input DataA
  - Variable Width of Input DataB
  - Output Pins Truncation
- Unsigned Multiplier, Pipeline x 1 Only
  - Variable Width of Input DataA
  - Variable Width of Input DataB
  - Initialization Polarity Selection
  - Initialization Selection

Programmable SLI

AT40K

AT40KAL

AT94K

Application Note

Rev. 2440A–12/01

Multiplier – Serial Parallel

The Serial-parallel Multiplier (SPM) generator can be used to create a signed or unsigned serial by parallel multiplier with serial output. A width *n* unsigned multiplier is constructed in the FPGA by stringing *n* Carry-Save Adders. A width *n* signed multiplier is constructed by stringing n-1 Carry-Save Adders and a two's complement together.

Let X and Y be the n-bit and m-bit number respectively. The parallel input X is multiplied by each bit of serial input with the least significant bit coming first, and each of those partial products is added to the shifted accumulation of the previous product. The serial output is then taken from the output of the least significant bit adder. The output bit has the same weight as the previous serial input bit, and the number of bits in the output is equal to the sum of the number of bits in each of the inputs.

Thus the product[1] is:

$$\{Y\_(m - 1)\} * \{X\_(n - 1) * 2^{(n - 1)} + X\_(n - 2) * 2^{(n - 2)} + ... + X\_(0)\}$$
$$+ \{Y\_(m - 2)\} * \{X\_(n - 1) * 2^{(n - 1)} + X\_(n - 2) * 2^{(n - 2)} + ... + X\_(0)\}$$
$$+ ...$$
$$+ \{Y\_(0)\} * \{X\_(n - 1) * 2^{(n - 1)} + X\_(n - 2) * 2^{(n - 2)} + ... + X\_(0)\}$$

Note: 1. The multiplier uses an asynchronous initialization scheme. After each multiplication operation is complete, the reset pin should be asserted for one clock cycle. This flushes the carry-save registers and is required for correct operation of the multiplier.

Parameters

Parameter	Value	Explanation
Width	Integer > 1	Width of parallel input data
Representation	Signed	Treat Inputs as signed
	Unsigned	Treat inputs as unsigned
Invert Clock	Boolean	Invert the clock input
Initialization Polarity = Low	Boolean	Reset input is active low

Pins

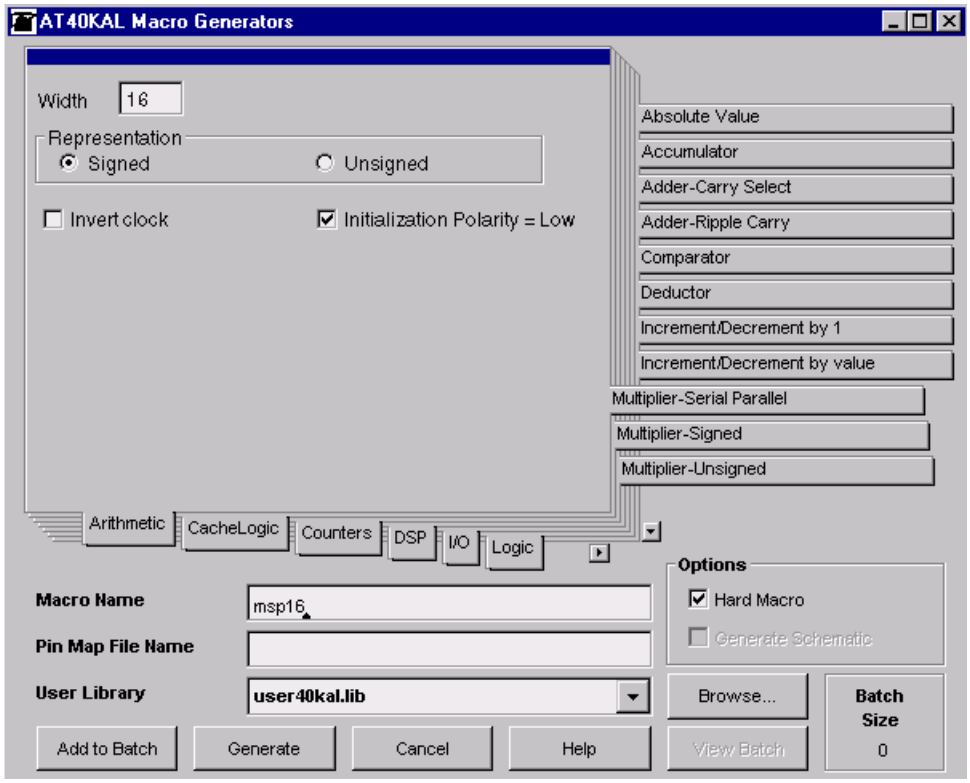
Type	Name	Option	Explanation
In	X[Width - 1:0]	No	Multiplier (parallel input)
In	Y	No	Multiplicand (serial input)
In	R/RN	No	Reset input (active high/low)
Out	PROD	No	Product of x and y (serial output)

Statistics

Device	Name	Speed (MHz)	Delay (ns)	Cells	Size (x * y)
AT40K	msp16	121.8	8.2	32	2 x 16
AT40K	msp8	121.8	8.2	16	2 x 8
AT94K/ AT40KAL	msp16	108.3	9.2	32	2 x 16
AT94K/ AT40KAL	msp8	108.3	9.2	16	2 x 8

Figure 1 shows an example of the msp16 macro options.

Figure 1. Multiplier – Serial Parallel Generator



Multiplier – Signed

The Signed-multiplier generator can be used to generate the product of two varying width inputs. The function it produces is the following:

Product = DATAA * DATAB

where DATAA and DATAB are treated as two's complement signed numbers.

Parameters

Parameter	Value	Explanation
WidthA	Integer > 2	Width of input DataA
WidthB	Integer > 2	Width of input DataB

Pins

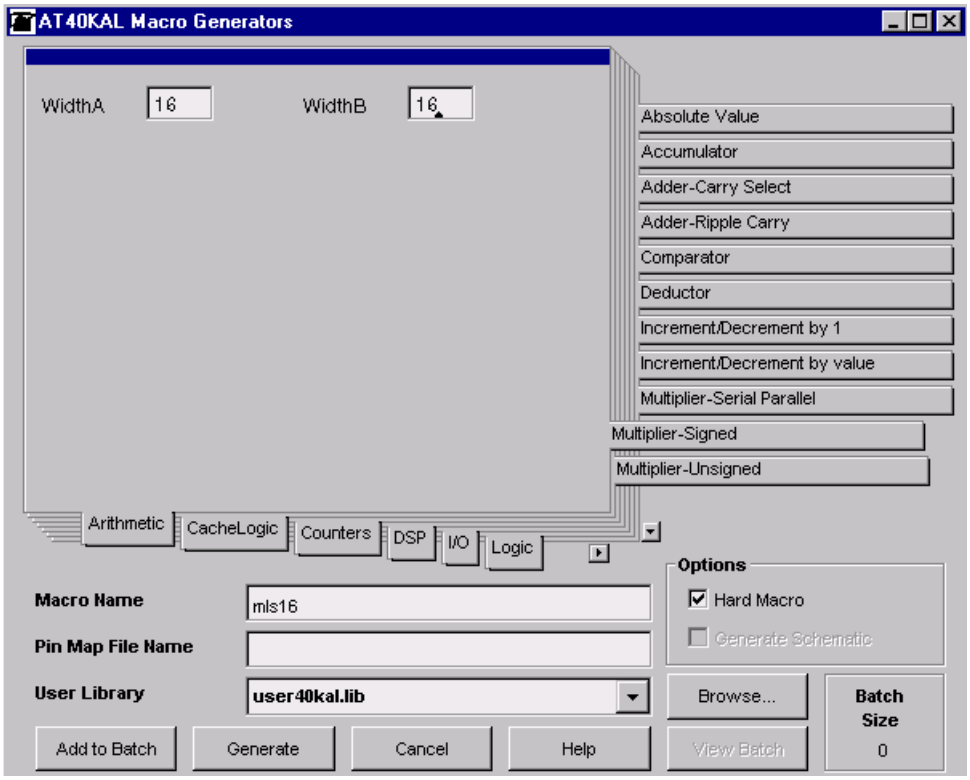
Type	Name	Option	Explanation
In	DATAA[WidthA - 1:0]	No	Multiplicand
In	DATAB[WidthB - 1:0]	No	Multiplier
Out	PRODUCT[Width - 1:0]	No	DataA * DataB (Width is WidthA + WidthB)

Statistics

Device	Name	Speed (MHz)	Delay (ns)	Cells	Size (x * y)
AT40K	mls16	15.5	64.3	289	17 x 17
AT40K	mls8	30.0	33.4	81	9 x 9
AT94K/ AT40KAL	mls16	22.2	45.1	289	17 x 17
AT94K/ AT40KAL	mls8	40.8	24.5	81	9 x 9

Figure 2 shows an example of the mls16 macro options.

Figure 2. Multiplier – Signed Generator



Multiplier – Signed, Pipeline x 1

This component can be used to generate the product of two varying width inputs. A single stage of pipelining is used to produce a considerably faster macro than the standard signed multiplier with minimal additional logic.

The function it produces is the following:

Product = DATAA * DATAB

where DATAA and DATAB are treated as two's complement signed numbers.

Parameters

Parameter	Value	Explanation
WidthA	Integer > 2	Width of input DataA
WidthB	Integer > 2	Width of input DataB
Invert Clock	Boolean	Invert the clock input
Initialization Polarity = Low	Boolean	Reset input is active low
Initialization	Reset	Provide a reset input for initialization of the pipeline registers
	None	Pipeline registers are automatically initialized on power-up

Pins

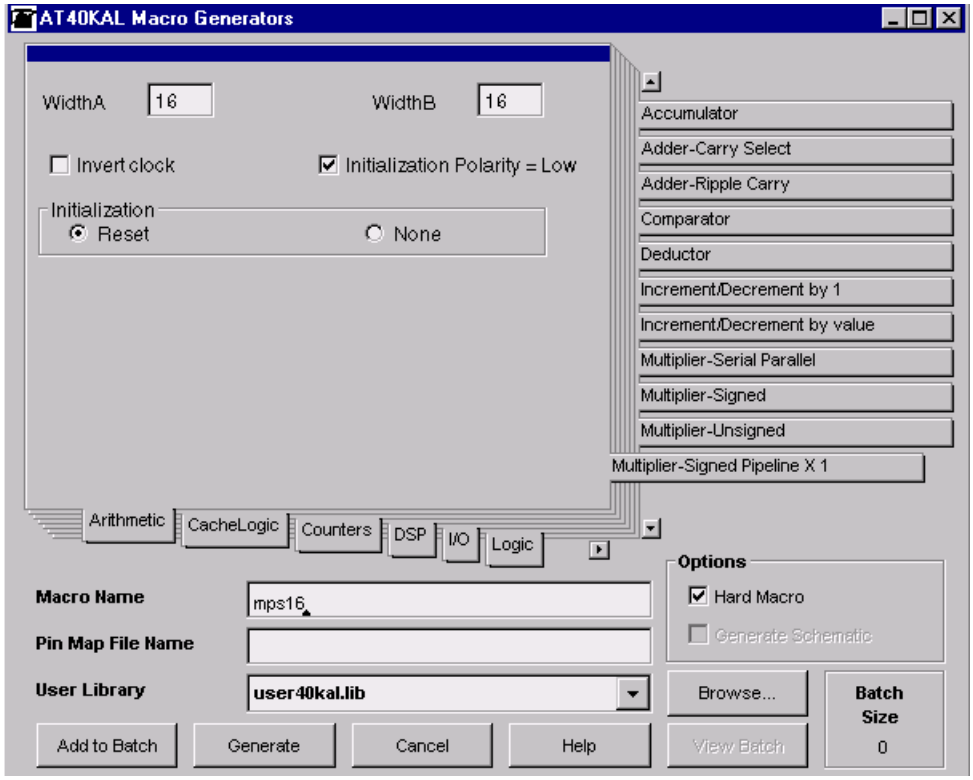
Type	Name	Option	Explanation
In	DATAA[WidthA - 1:0]	No	Multiplicand
In	DATAB[WidthB - 1:0]	No	Multiplier
In	CLK/CLKN	No	Clock input (noninverted/inverted)
In	R/RN	Yes	Reset input (active high/low)
Out	PRODUCT[Width - 1:0]	No	DataA * DataB (Width is WidthA + WidthB)

Statistics

Device	Name	Speed (MHz)	Delay (ns)	Cells	Size (x * y)
AT40K	mps16	24.9	40.1	306	17 x 18
AT40K	mps8	44.7	22.4	90	9 x 10
AT94K/ AT40KAL	mps16	33.8	29.6	306	17 x 18
AT94K/ AT40KAL	mps8	68.6	14.6	90	9 x 10

Figure 3 shows an example of the mps16 macro options.

Figure 3. Multiplier – Signed Pipeline x 1 Generator



Multiplier – Unsigned

The Unsigned-multiplier generator can be used to generate the product of two varying width inputs. The function it produces is the following:

Product = DATAA * DATAB

Where DATAA and DATAB are treated as unsigned numbers.

Parameters

Parameter	Value	Explanation
WidthA	Integer > 2	Width of input DataA
WidthB	Integer > 2	Width of input DataB
Truncate Result to n Least Significant Bits	Integer ≥ 0	The number of output pins that have to be included in the layout starting from the LSB. This option saves the layout area by not including the unnecessary output pins. If this value is left at 0, all outputs will be present.

Pins

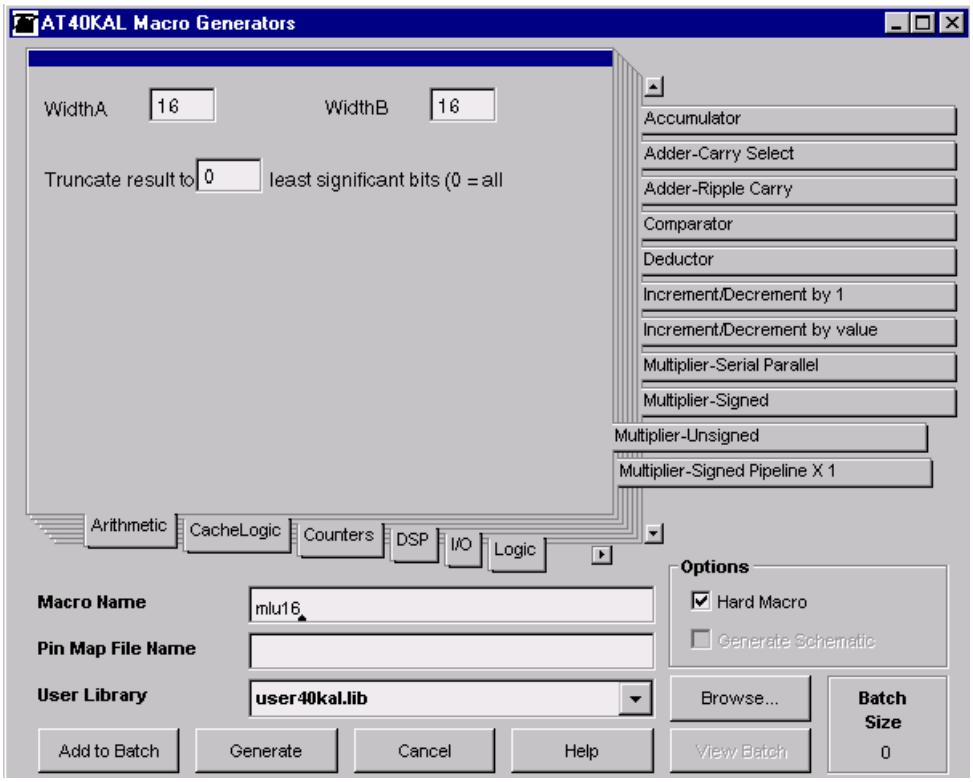
Type	Name	Option	Explanation
In	DataA[WidthA - 1:0]	No	Multiplicand
In	DataB[WidthB - 1:0]	No	Multiplier
Out	PRODUCT[Width - 1:0]	No	DataA * DataB (Width is WidthA + WidthB)

Statistics

Device	Name	Speed (MHz)	Delay (ns)	Cells	Size (x * y)
AT40K	mlu16	16.0	62.7	256	16 x 16
AT40K	mlu8	31.5	31.7	64	8 x 8
AT94K/ AT40KAL	mlu16	22.8	43.8	256	16 x 16
AT94K/ AT40KAL	mlu8	42.9	23.3	64	8 x 8

Figure 4 shows an example of the mlu16 macro options.

Figure 4. Multiplier – Unsigned Generator



Multiplier – Unsigned, Pipeline x 1

This component can be used to generate the product of two varying width inputs. A single stage of pipelining is used to produce a considerably faster macro than the standard unsigned multiplier with minimal additional logic.

The function it produces is the following:

Product = DATAA * DATAB

Where DATAA and DATAB are treated as unsigned numbers.

Parameters

Parameter	Value	Explanation
WidthA	Integer > 2	Width of input DataA
WidthB	Integer > 2	Width of input DataB
Invert Clock	Boolean	Invert the clock input
Initialization Polarity = Low	Boolean	Reset input is active low
Initialization	Reset	Provide a Reset Input for Initialization of the Pipeline Registers
	None	Pipeline Registers are Automatically Initialized on Power-up

Pins

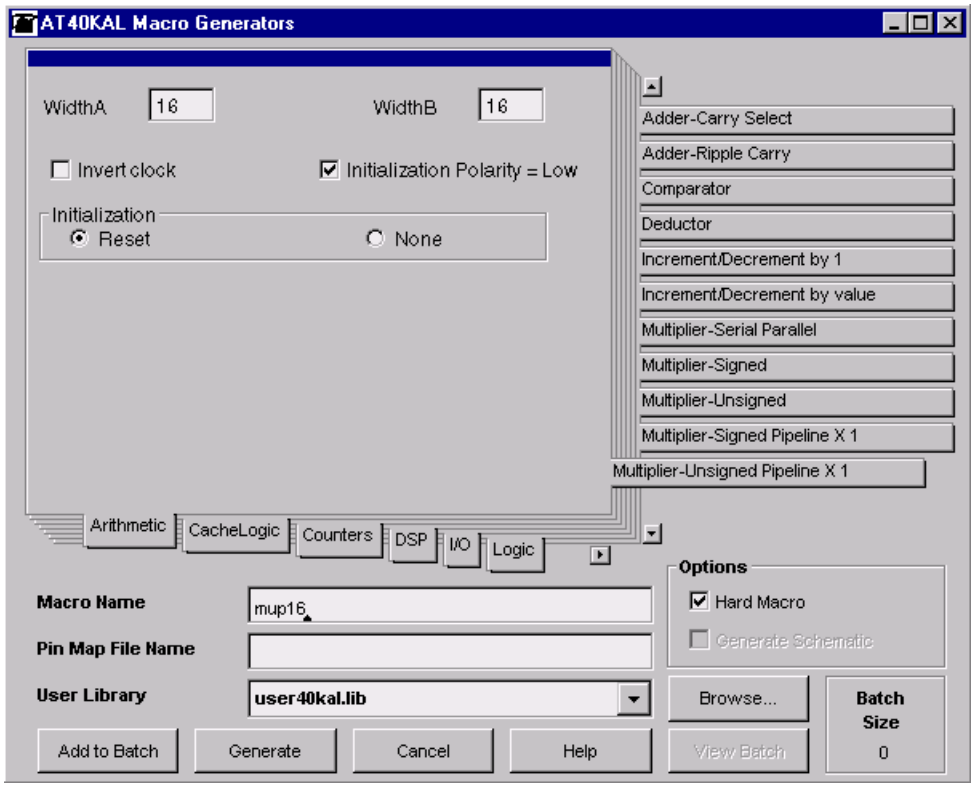
Type	Name	Option	Explanation
In	DATAA[WidthA - 1:0]	No	Multiplicand
In	DATAB[WidthB - 1:0]	No	Multiplier
In	CLK/CLKN	No	Clock input (noninverted/inverted)
In	R/RN	Yes	Reset input (active high/low)
Out	PRODUCT[Width - 1:0]	No	DataA * DataB (Width is WidthA + WidthB)

Statistics

Device	Name	Speed (MHz)	Delay (ns)	Cells	Size (x * y)
AT40K	mup16	27.3	36.7	272	16 x 17
AT40K	mup8	52.9	18.9	72	8 x 9
AT94K/ AT40KAL	mup16	38.7	25.8	272	16 x 17
AT94K/ AT40KAL	mup8	84.9	11.8	72	8 x 9

Figure 5 shows an example of the mup16 macro options.

Figure 5. Multiplier – Unsigned Generator



IP Core Generator: Mux

Features

- **Accessible from the Macro Generator Dialog and HDLPlanner™ – Included in IDS for FPGA Devices and System Designer™ for AT94K FPSLIC™ Devices**
- **Variable Number of Parallel Muxes**
- **Variable Number of Inputs per Mux**

Description

The Mux generator can be used to generate one or more muxes controlled by a single SELECT bus input. The width of each Mux is user programmable.

Parameters

Parameter	Value	Explanation
Number of Muxes	Integer ≥ 1	Number of parallel mux components in the macro
Number of Inputs per Mux	Integer ≥ 1	Number of inputs that each mux switches

Pins(1)

Type	Name	Option	Explanation
In	S[log2 (Width) - 1:0]	No	Mux select pins, Log2-bits wide
In	DATA[Size - 1:0]	No	Mux input pins
Out	RESULT[Size - 1:0]	No	Mux output

Note: 1. Size represents the number of muxes and Width is the number of inputs to each mux.

Programmable SLI
AT40K
AT40KAL
AT94K

Application Note

Rev. 2441A–12/01

Truth Table

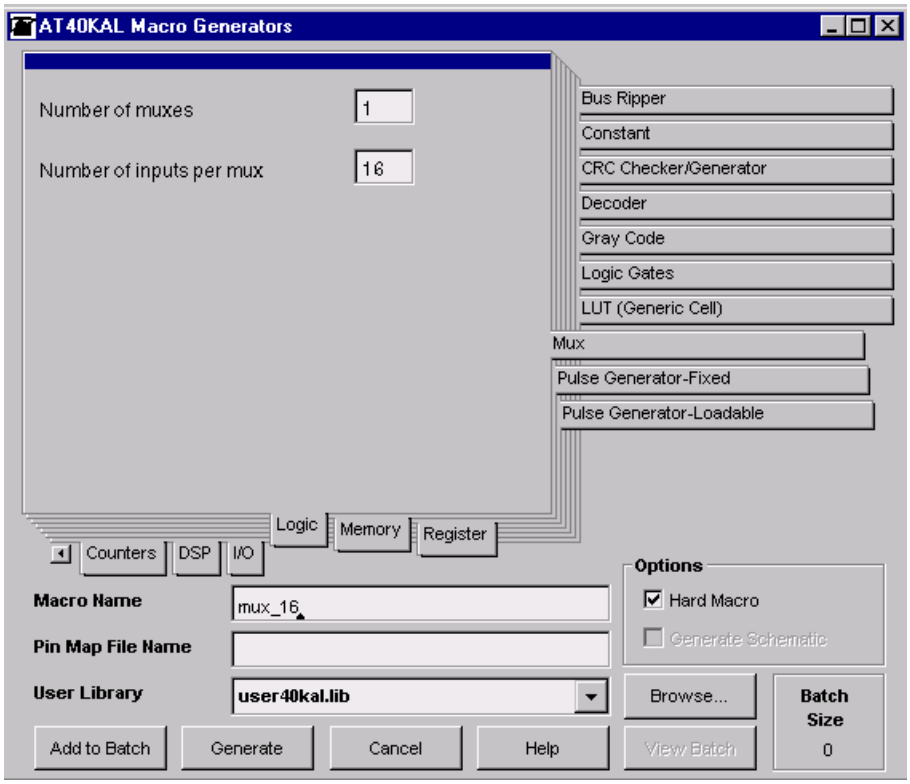
Input		Output
S	DATA	RESULT
0	..XXX0	0
0	..XXX1	1
1	..XX0X	0
1	..XX1X	1
2	..X0XX	0
2	..X1XX	1
. . .		. . .

Statistics

Device	Name	Speed (MHz)	Delay (ns)	Cells	Size (x * y)
AT40K	mux_16	127.1	7.9	16	8 x 2
AT40K	mux_8	130.5	7.7	8	4 x 2
AT94K/ AT40KAL	mux_16	140.6	7.1	16	8 x 2
AT94K/ AT40KAL	mux_8	154.1	6.5	8	4 x 2

Figure 1 shows an example of the mux_16 macro options.

Figure 1. Mux Generator



IP Core Generator: Negate Function

Features

- **Accessible from the Macro Generator Dialog and HDLPlanner™ – Included in IDS for FPGA Devices and System Designer™ for AT94K FPSLIC™ Devices**
- **Optional Overflow Output Pin**
- **Variable Width of Input and Output Data**
- **Variable Number of Inputs per Mux**

Description

The Negate Function generator can be used to create a two's complement function implemented in a ripple carry manner.

if DATA = $2^{(Width - 1)}$, then

OVERFLOW = 1, RESULT = -DATA

else

RESULT = -DATA

DATA must always represent a positive number and the RESULT is always a two's complement number.

Parameters

Parameter	Value	Explanation
Overflow	Boolean	Overflow output pin is present
Width	Integer > 1	Width of input and output data

Pins

Type	Name	Option	Explanation
In	DATA[Width - 1:0]	No	Data input
Out	RESULT	No	Data output = two's complement of data
Out	OVERFLOW	Yes	1 if data = $2^{(Width - 1)}$, 0 otherwise

Programmable SLI
AT40K
AT40KAL
AT94K

Application Note

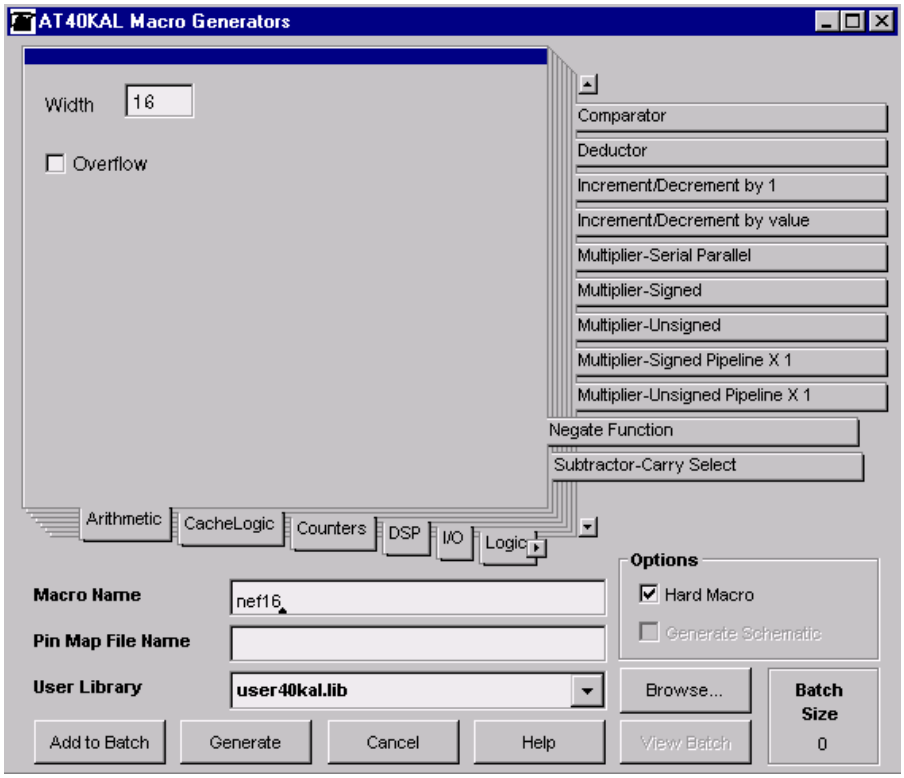
Rev. 2442A–12/01

Statistics

Device	Name	Speed (MHz)	Delay (ns)	Cells	Size (x * y)
AT40K	nef16	36.4	27.4	16	1 x 16
AT40K	nef8	70.2	14.2	8	1 x 8
AT94K/ AT40KAL	nef16	51.4	19.4	16	1 x 16
AT94K/ AT40KAL	nef8	101.6	9.8	8	1 x 8

Figure 1 shows an example of the nef16 macro options.

Figure 1. Negate Function Generator



IP Core Generator: Pulse Generator

Features

- **Pulse Generator – Fixed**
- **Pulse Generator – Loadable**
- **Accessible from the Macro Generator Dialog and HDLPlanner™ – Included in IDS for FPGA Devices and System Designer™ for AT94K FPSLIC™ Devices**
- **Variable Pitch**
- **Clock Inversion Capability**
- **Initialization Polarity Selection**
- **Fixed Pulse Generator Only**
  - **Variable Output Clock Cycles**
  - **Initialization Radix of Above Value Selection**
- **Loadable Pulse Generator Only**
- **Variable Width of Pulse Generator**

Pulse Generator – Fixed

This can be used to create a pulse generator that asserts its output once every *n* clock cycles, where *n* is a fixed value specified by the user.

Parameters

Parameter	Value	Explanation
Generate a Pulse Every *n* Clock Cycles	Integer > 1	Output will be low for *n* - 1 clock cycle, then high for 1 clock cycle, in a repeating pattern
Radix of Above Value	Binary	*n* is specified in binary representation
	Octal	*n* is specified in octal representation
	Decimal	*n* is specified in decimal representation
	Hex	*n* is specified in hexadecimal representation
Pitch	Integer ≥ 1	Spacing between cells in the pulse generator. A pitch of 2 doubles the size of the generator by spreading out its layout.
Invert Clock	Boolean	Inverts the clock input
Initialization Polarity = Low	Boolean	Preset input is active low

Programmable SLI
AT40K
AT40KAL
AT94K

Application Note

Rev. 2443A–12/01

Pins

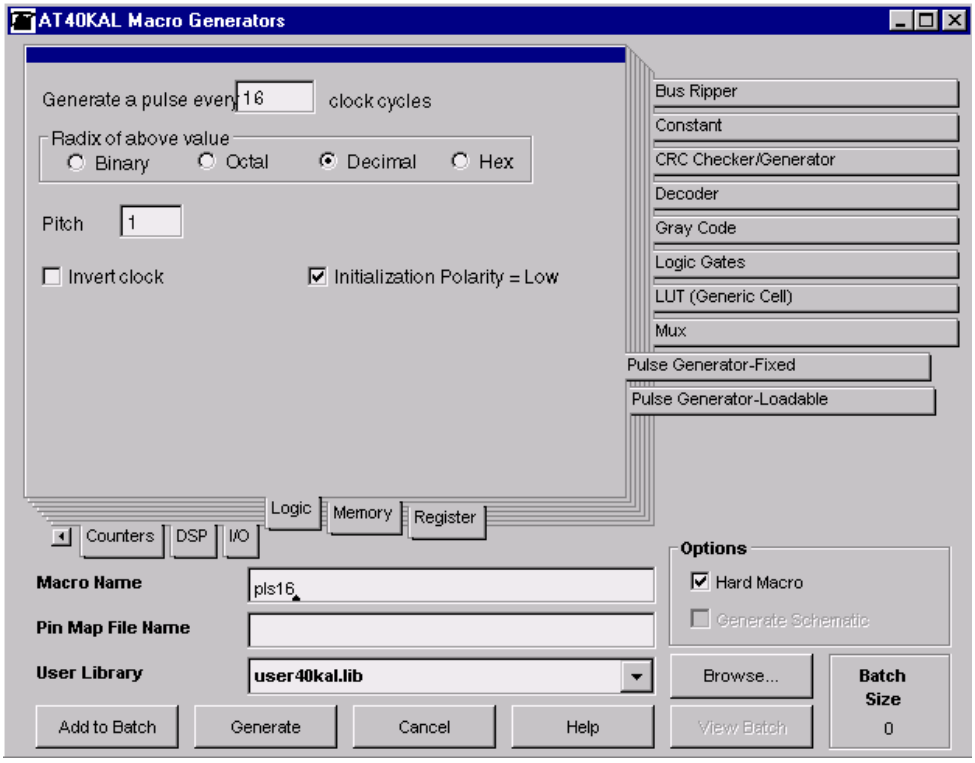
Type	Name	Option	Explanation
In	CLK/CLKN	No	Clock (noninverted/inverted)
In	P/PN	No	Preset to starting value (active high/low)
Out	TERMCNT	No	Terminal count (pulse output)

Statistics

Device	Name	Speed (MHz)	Delay (ns)	Cells	Size (x * y)
AT40K	pls16	133.7	7.5	5	1 x 5
AT40K	pls8	171.5	5.8	4	1 x 4
AT94K/ AT40KAL	pls16	141.4	7.1	5	1 x 5
AT94K/ AT40KAL	pls8	171.5	5.8	4	1 x 4

Figure 1 shows an example of the pls16 macro options.

Figure 1. Pulse Generator – Fixed



Pulse Generator – Loadable

This can be used to create a pulse generator that asserts its output once every *n* clock cycles, where *n* is a value that is loaded into the macro at run-time. Performing a parallel load operation modifies the value of *n*, i.e., the pulse frequency.

Parameters

Parameter	Value	Explanation
Width	Integer > 1	Width of the pulse generator (i.e., the number of registers it contains). This parameter dictates the maximum size of *n* that can be loaded into the macro.
Pitch	Integer ≥ 1	Spacing between cells in the pulse generator. A pitch of 2 doubles the size of the generator by spreading out its layout.
Invert Clock	Boolean	Inverts the clock input
Initialization Polarity = Low	Boolean	Reset input is active low

Pins

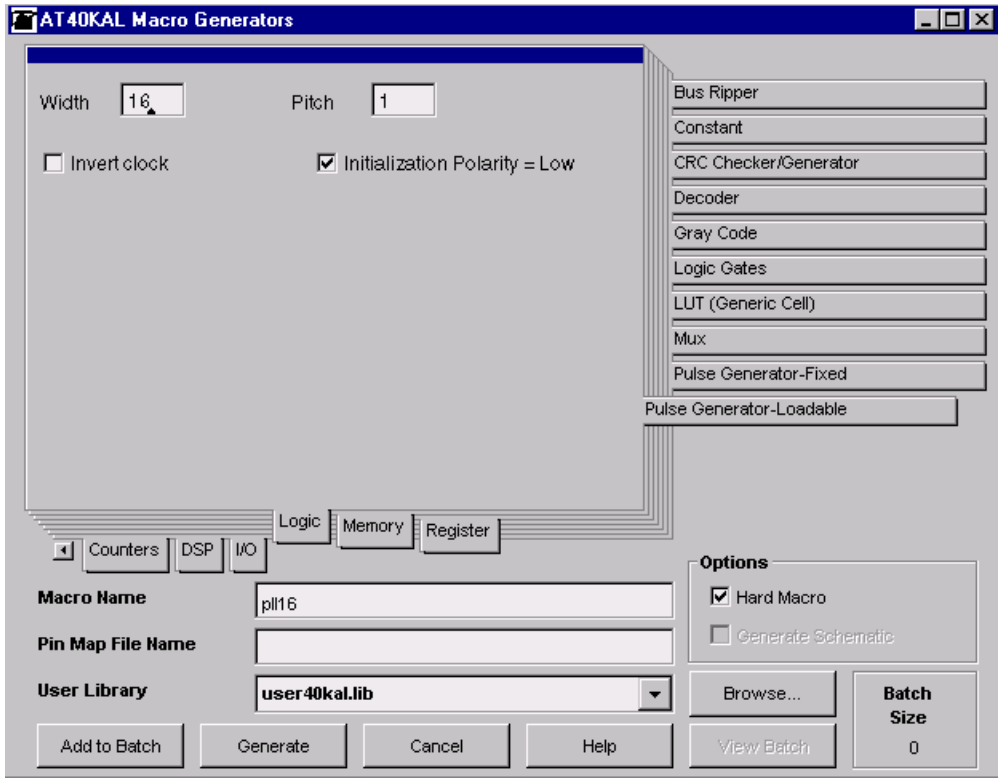
Type	Name	Option	Explanation
In	SLOAD	No	0 = Load pulse frequency value; 1 = Generate pulses
In	DATA[Width - 1:0]	No	Parallel load inputs for pulse frequency value
In	CLK/CLKN	No	Clock (noninverted/inverted)
In	R/RN	No	Reset (active high/low)
Out	TERMCNT	No	Terminal count (pulse output)

Statistics

Device	Name	Speed (MHz)	Delay (ns)	Cells	Size (x * y)
AT40K	pll16	20.3	49.3	48	3 x 17
AT40K	pll8	38.6	25.9	24	3 x 9
AT94K/ AT40KAL	pll16	25.2	39.8	48	3 x 17
AT94K/ AT40KAL	pll8	47.9	20.9	24	3 x 9

Figure 2 shows an example of the pll16 macro options.

Figure 2. Pulse Generator – Loadable



IP Core Generator: ROM

Features

- **Accessible from the Macro Generator Dialog and HDLPlanner™ – Included in IDS for FPGA Devices and System Designer™ for AT94K FPSLIC™ Devices**
- **Variable Width of Address Inputs**
- **Variable Width of ROM Data**
- **ROM Filename Selection**
- **Variable Enable Input**

Description

The ROM generator is used to create synchronous read-only data memories. The data is generated by reading a user-supplied data file and then using the look-up tables within each core cell to store the data.

Parameters

Parameter	Value	Explanation
Address Width	Integer > 0	Width of address inputs – the number of ROM words created is determined by this parameter, e.g. 2adwidth
Width	Integer > 0	Width of each ROM data word
ROM File	File name	Name of the file with data to be stored in the ROM
Enable Input	Boolean	Provides an enable input (active high)

Pins

Type	Name	Option	Explanation
In	ADDRESS[AD - Width - 1:0]	No	Input address for ROM
In	OE	Yes	Memory enable 1 = output data; 0 = tri-state
Out	Q[Width - 1:0]	No	ROM data output

Programmable SLI
AT40K
AT40KAL
AT94K

Application Note

Rev. 2445A–12/01

The ROM generator reads as input a file specified by the file option (default is rom.hex) which should be located in the project directory. The format of the file is basically a header line with either hex, dec, oct or bin to indicate that numbers in the file are in hexadecimal, decimal, octal or binary format, respectively. Each subsequent line should be of the form Address Value. For example:

rom.hex(1)

```
dec
0   0
1   2
2   4
3   8
4   7
5   5
6   3
7   3
8   3
9   3
10  3
11  3
12  4
13  5
14  6
15  7
```

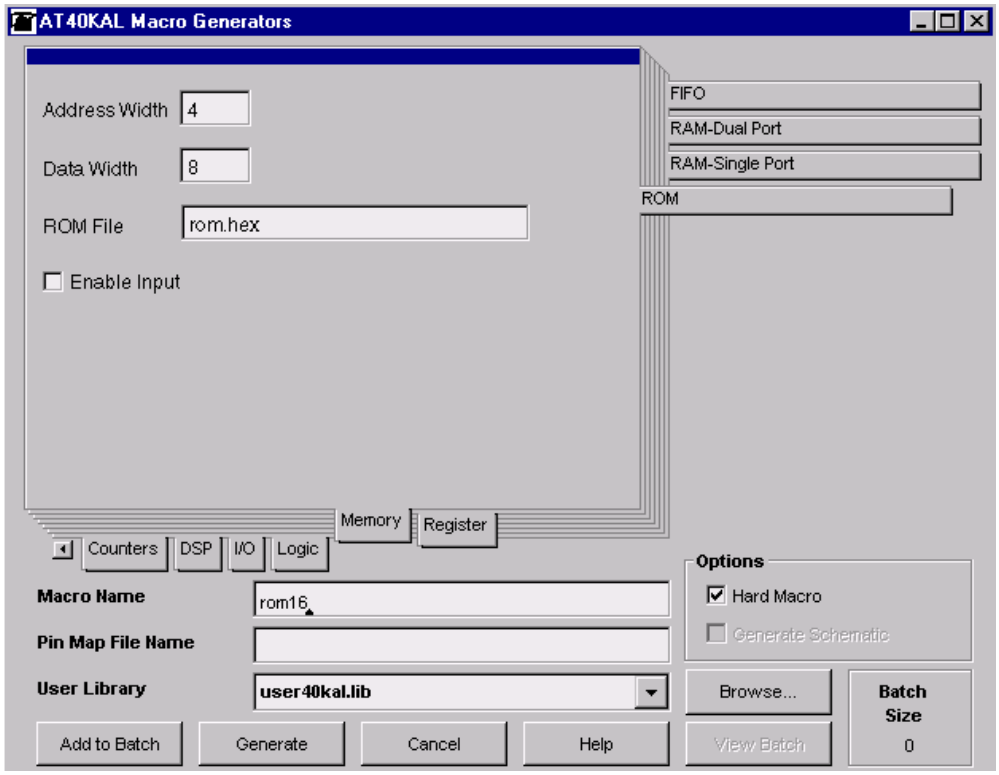
Note: 1. The left column refers to the address line, and the right column refers to the data line.

Statistics

Device	Name	Speed (MHz)	Delay (ns)	Cells	Size (x * y)
AT40K	rom16	339.0	3.0	8	8 x 1
AT40K	rom8	339.0	3.0	4	4 x 1
AT94K/ AT40KAL	rom16	492.6	2.0	8	8 x 1
AT94K/ AT40KAL	rom8	492.6	2.0	4	4 x 1

Figure 1 shows an example of the rom16 macro options.

Figure 1. ROM Generator



IP Core Generator: Shift Register

Features

- **Accessible from the Macro Generator Dialog and HDLPlanner™ – Included in IDS for FPGA Devices and System Designer™ for AT94K FPSLIC™ Devices**
- **Input Type Selection**
- **Output Type Selection**
- **Variable Width of Shift Register**
- **Optional Enable**
- **Clock Inversion Capability**
- **Initialization Polarity Selection**
- **Initialization Selection**
- **Preset and Parallel Input Radix Selection**

Description

The Shift-register generator can be used to create serial, serial-parallel, parallel-serial and parallel-parallel shift registers.

Programmable SLI
AT40K
AT40KAL
AT94K

Application Note

Rev. 2446A–12/01

Parameters

Parameter	Value	Explanation
Input Type	Serial	Input is serial
	Parallel (Variable)	Input is parallel
	Parallel (Constant)	Input is parallel, with a fixed value
Output Type	Serial	Output is serial
	Parallel	Output is parallel
Width	Integer > 0	Width of shift register
Enable	Boolean	Add an enable pin to the register
Invert Clock	Boolean	Invert the clock input
Initialization polarity = Low	Boolean	Reset/set/preset input is active low
Initialization	Reset	Registers can be reset to zero
	Set	Registers can be set to one
	None	Registers are reset automatically on power-up
	Preset	Registers can be asynchronously loaded with a constant value
Preset and Parallel Input Radix	Binary	Preset and parallel input values are specified using binary representation
	Octal	Values are specified in octal
	Decimal	Values are specified in decimal
	Hex	Values are specified in hexadecimal

Pins

Type	Name	Option	Explanation
In	DATA[Width - 1:0]	Yes	Data input for parallel-input shift registers
In	SHIFTIN	Yes	Data input for serial-input shift registers
In	CLK/CLKN	No	Clock (noninverted/inverted)
In	ENABLE	Yes	Register enable input
In	SHIFTEN	Yes[1]	0 = load data in 1 = shift
In	R/RN/S/SN/P/PN	No	Reset/set/preset (active high/low)
Out	Q[Width - 1:0]	Yes	Data output for parallel-output shift registers
Out	SHIFTOUT	Yes	Data output for serial-output shift registers

Note: 1. With parallel input

Truth Table[1]

Input						Output	
RN	DATA[W - 1:0]	CLK	ENABLE	SHIFTIN	SHIFTEN	Q[W - 1:0]	SHIFTOUT
0	X	X	X	X	X	0	Q(N - 1)
1	X	0 >1	X	X	X	No Change	Q(N - 1)
1	0	0 >1	1	X	1	0	Q(N - 1)
1	1	0 >1	1	X	1	1	Q(N - 1)
1	X	0 >1	1	SI	0	Q(I) = Q- (I - 1)[2] Q(0) = SI	Q(N - 2)

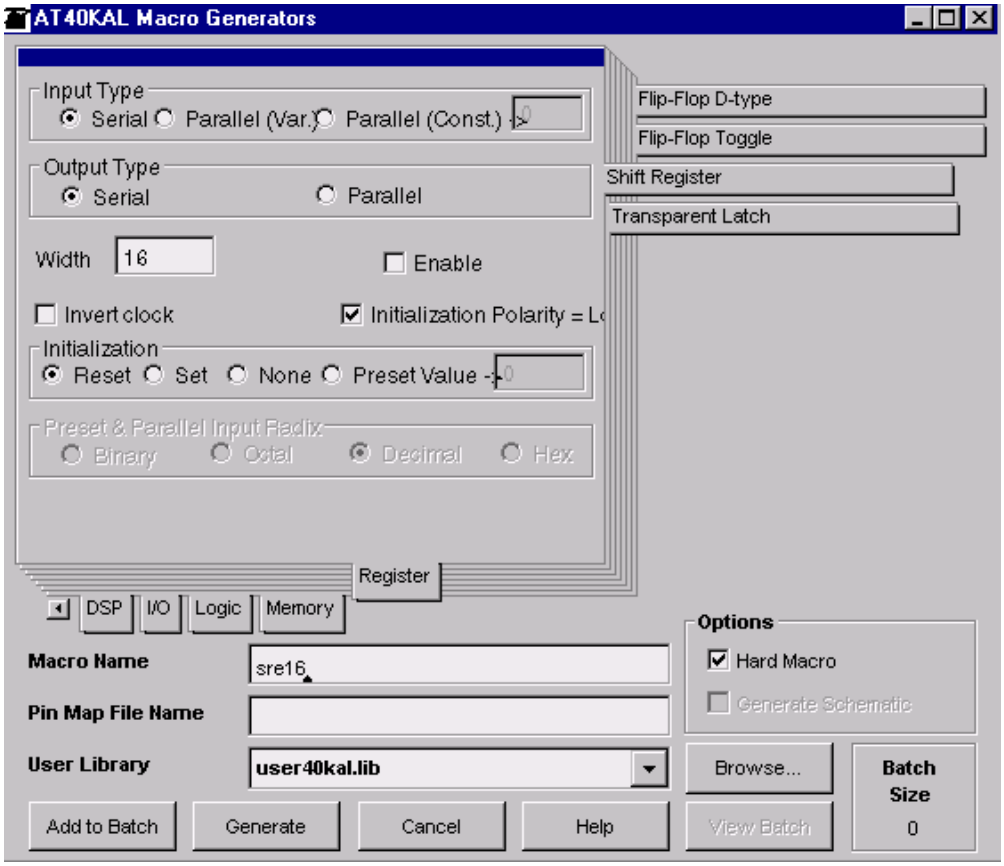
Notes: 1. This truth table assumes that an active-low reset and non-inverted clock have been selected.
2. Q - is the value of Q preceding the clock transition.

Statistics

Device	Name	Speed (MHz)	Delay (ns)	Cells	Size (x * y)
AT40K	sre16	226.2	4.4	16	1 x 16
AT40K	sre8	226.2	4.4	8	1 x 8
AT94K/ AT40KAL	sre16	598.8	1.7	16	1 x 16
AT94K/ AT40KAL	sre8	598.8	1.7	8	1 x 8

Figure 1 shows an example of the sre16 macro options.

Figure 1. Shift Register



IP Core Generator: Subtractor

Features

- **Subtractor – Carry Select**
- **Subtractor – Ripple Carry**
- **Accessible from the Macro Generator Dialog and HDLPlanner™ – Included in IDS for FPGA Devices and System Designer™ for AT94K FPSLIC™ Devices**
- **Carry-in Pin Option**
- **Carry-out Pin Option**
- **Variable Width of Input and Output Vectors**
- **Ripple-carry Subtractor Only**
  - **Register Inputs and Outputs Selection**
  - **Optional Signed Overflow Pin**
  - **Variable Pitch**
  - **Variable Aspect Ratio**

Subtractor – Carry Select

This generator can be used to create an *n* bit carry-select subtractor.

Parameters

Parameter	Value	Explanation
Width	Integer > 1	Width of input and output vectors
Carry In	Boolean	Provides a carry-in pin
Carry Out	Boolean	Provides a carry-out pin

Pins

Type	Name	Option	Explanation
In	CIN	Yes	Carry in
In	DATAA[Width - 1:0]	No	A input
In	DATAB[Width - 1:0]	No	B input
Out	SUM[Width - 1:0]	No	Subtractor output
Out	COUT	Yes	Carry out

Programmable SLI
AT40K
AT40KAL
AT94K

Application Note

Rev. 2447A–12/01

Truth Table

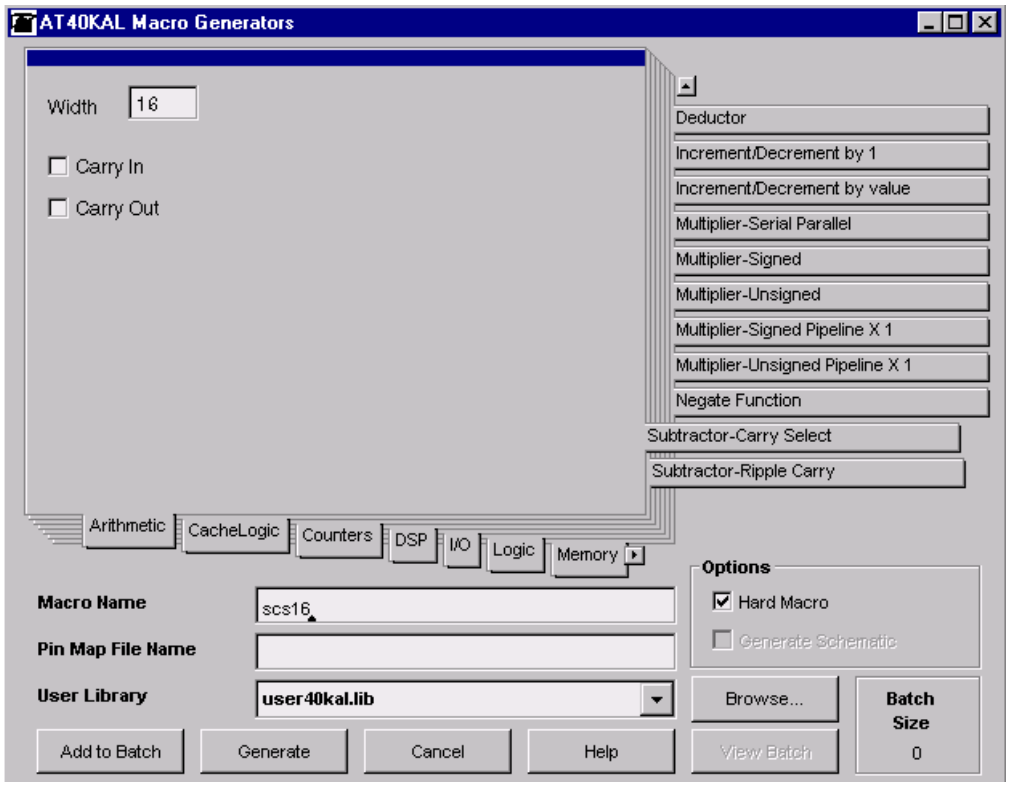
Input			Output	
CIN	DATAA[W - 1:0]	DATAB[W - 1:0]	SUM[W - 1:0]	COUT
C	A	B	A - B - C	1 if A - B - C < 0, 0 otherwise

Statistics

Device	Name	Speed (MHz)	Delay (ns)	Cells	Size (x * y)
AT40K	scs16	40.0	25.0	48	3 x 19
AT94K/ AT40KAL	scs16	47.0	21.3	48	3 x 19

Figure 1 shows an example of the scs16 macro options.

Figure 1. Subtractor – Carry Select Generator



Subtractor – Ripple Carry

This generator can be used to create a Ripple-carry Subtractor.

Parameters

Parameter	Value	Explanation
CarryIn	NoRegister	Includes the carry-in pin on the generator, but does not register it
	Register	Registers carry-in of the subtractor
	Disabled	Does not include the carry-in pin on the subtractor
CarryOut	NoRegister	Includes the carry-out pin on the subtractor but does not register it
	Register	Registers carry-out of the subtractor
	Disabled	Does not include the carry-out pin on the subtractor
Register	None	Does not register the inputs and outputs
	Input	Registers inputs on the subtractor, and excludes carry-in pin
	Output	Registers outputs on the subtractor, and excludes carry-out pin
	Both	Registers both inputs and outputs including the carry-in and carry-out pins of the subtractor
Signed Overflow Pin	Boolean	Provides a signed overflow output (treating input vectors as signed values)
Pitch	Integer > 1	Spacing between input pins, pitch of 2 means one cell between input pins.
Width	Integer > 1	Width of input and output vectors
Aspect Ratio	Float ≥ 0.0	Aspect ratio of the subtractor layout. A ratio of 0:0 gives a thin, vertical layout; whereas a ratio of 1:0 gives a square layout.

If input, output, carry-in or carry-out registers are selected, three additional parameters are available.

Register Parameters

Parameter	Value	Explanation
Invert Clock	Boolean	Inverts the register clock
Initialization Polarity = Low	Boolean	Makes register initialization active low
Register Set/ Reset Function	Reset	Registers can be reset to zero
	Set	Registers can be set to one

Pins

Type	Name	Option	Explanation
In	CIN	Yes	Carry in
In	DATAA[Width - 1:0]	No	A input
In	DATAB[Width - 1:0]	No	B input
In	CLK/CLKN	Yes	Clock (noninverted/inverted)
In	R/RN/S/SN	Yes	Reset/set (active high/low)
Out	SUM[Width - 1:0]	No	Adder output
Out	COUT	Yes	Carry out (cannot be used with overflow in an unsigned adder)
Out	OVERFLOW	Yes	Overflow

Carry out = DATAA - DATAB - CIN > 2^n - 1 or

DATAA - DATAB - CIN < -2^n

Overflow for Unsigned

DATAA - DATAB - CIN > 2^n - 1 or

DATAA - DATAB - CIN < 0

Overflow for Signed

DATAA - DATAB - CIN > 2^n - 1 or

DATAA - DATAB - CIN < -2^n

Truth Table

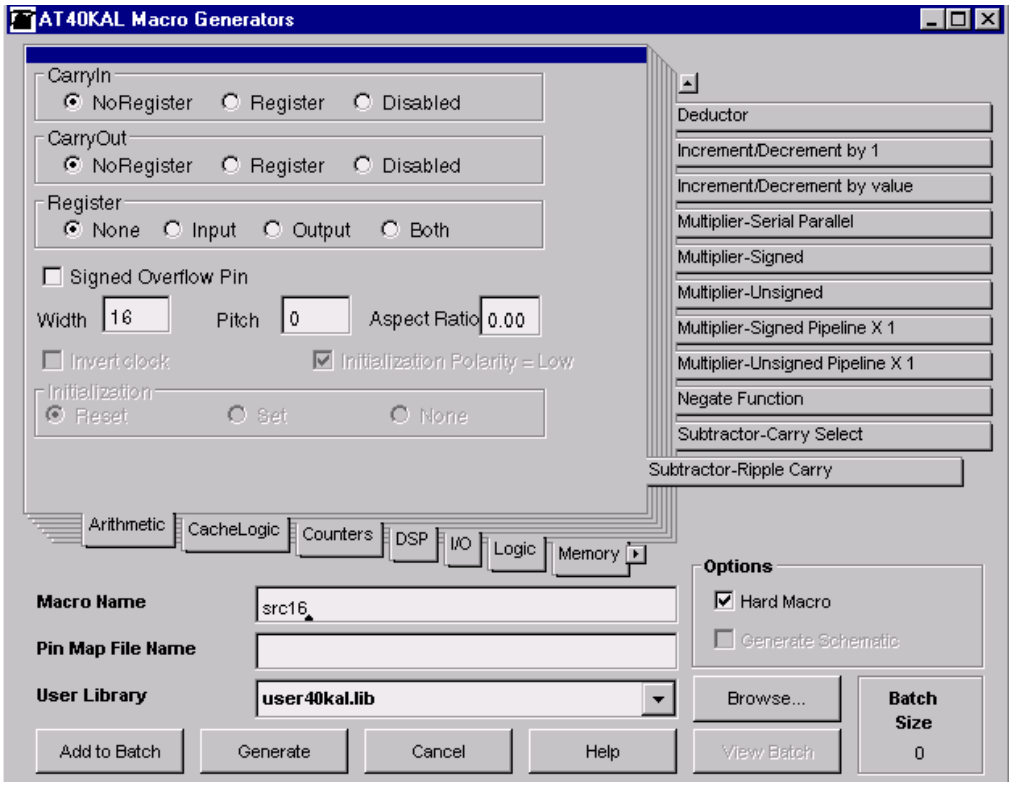
Input			Output	
CIN	DATAA[W - 1:0]	DATAB[W - 1:0]	SUM[W - 1:0]	COUT
C	A	B	A - B - C	1 if A - B - C < 0, 0 otherwise

Statistics

Device	Name	Speed (MHz)	Delay (ns)	Cells	Size (x * y)
AT40K	src16	36.2	27.7	16	1 x 16
AT40K	src8	69.2	14.5	8	1 x 8
AT94K/ AT40KAL	src16	49.9	20.1	16	1 x 16
AT94K/ AT40KAL	src8	95.6	10.5	8	1 x 8

Figure 2 shows an example of the src16 macro options.

Figure 2. Subtractor – Ripple Carry Generator



Atmel Headquarters

Corporate Headquarters
2325 Orchard Parkway
San Jose, CA 95131
TEL (408) 441-0311
FAX (408) 487-2600

Europe
Atmel SarL
Route des Arsenaux 41
Casa Postale 80
CH-1705 Fribourg
Switzerland
TEL (41) 26-426-5555
FAX (41) 26-426-5500

Asia
Atmel Asia, Ltd.
Room 1219
Chinachem Golden Plaza
77 Mody Road Tsimhatsui
East Kowloon
Hong Kong
TEL (852) 2721-9778
FAX (852) 2722-1369

Japan
Atmel Japan K.K.
9F, Tonetsu Shinkawa Bldg.
1-24-8 Shinkawa
Chuo-ku, Tokyo 104-0033
Japan
TEL (81) 3-3523-3551
FAX (81) 3-3523-7581

Atmel Product Operations

Atmel Colorado Springs
1150 E. Cheyenne Mtn. Blvd.
Colorado Springs, CO 80906
TEL (719) 576-3300
FAX (719) 540-1759

Atmel Grenoble
Avenue de Rochepleine
BP 123
38521 Saint-Egreve Cedex, France
TEL (33) 4-7658-3000
FAX (33) 4-7658-3480

Atmel Heilbronn
Theresienstrasse 2
POB 3535
D-74025 Heilbronn, Germany
TEL (49) 71 31 67 25 94
FAX (49) 71 31 67 24 23

Atmel Nantes
La Chantrerie
BP 70602
44306 Nantes Cedex 3, France
TEL (33) 0 2 40 18 18 18
FAX (33) 0 2 40 18 19 60

Atmel Rousset
Zone Industrielle
13106 Rousset Cedex, France
TEL (33) 4-4253-6000
FAX (33) 4-4253-6001

Atmel Smart Card ICs
Scottish Enterprise Technology Park
East Kilbride, Scotland G75 0QR
TEL (44) 1355-357-000
FAX (44) 1355-242-743

Atmel Programmable SLI Hotline
(408) 436-4119

Atmel Programmable SLI e-mail
fpga@atmel.com – fpslic@atmel.com

FAQ
Available on web site

e-mail
literature@atmel.com

Web Site
http://www.atmel.com

2447A–12/01/xM