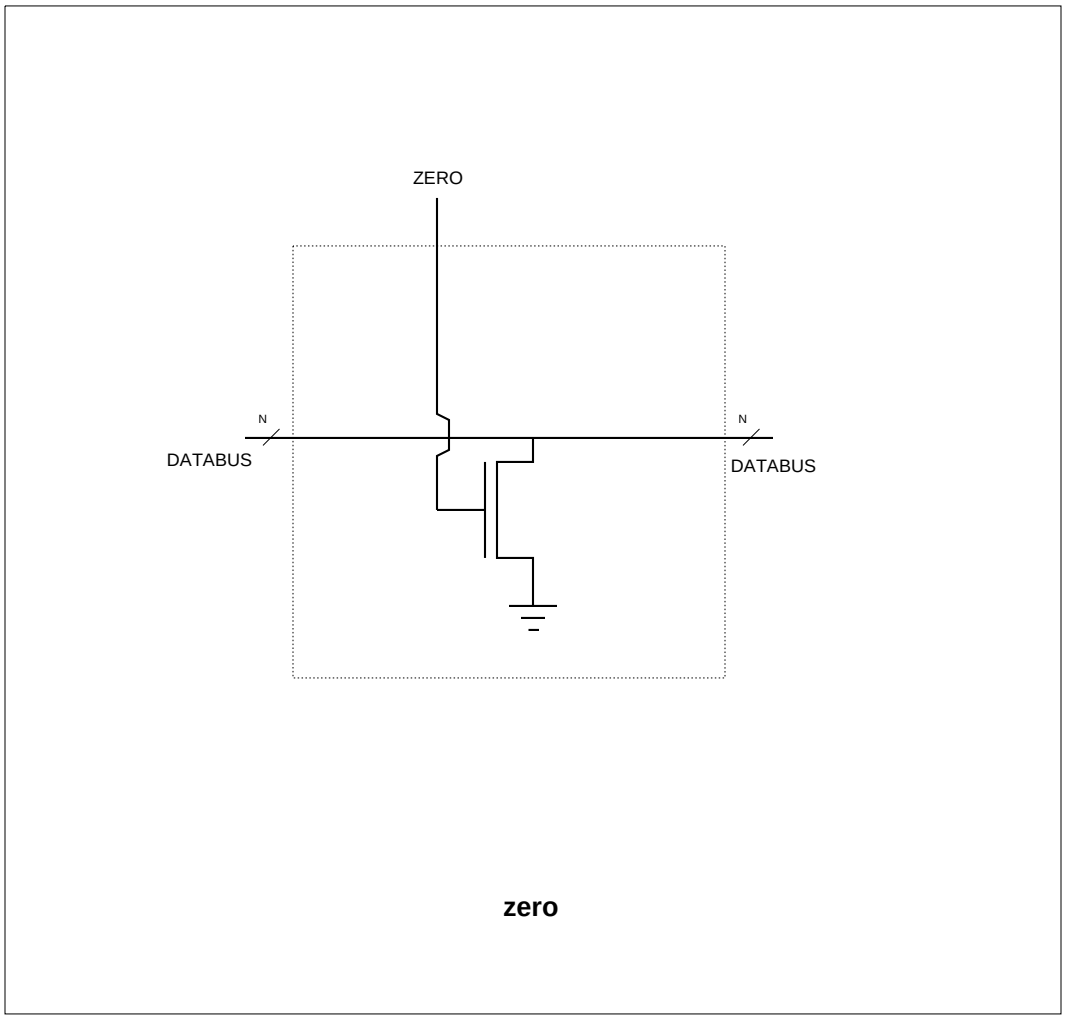

ZERO
N
DATABUS
N
DATABUS
zero

28 zero

Synopsis

This block can be used to selectively pull down a data bus. When the CONTROL signal ZERO is asserted to 1, the N-bit wide bus DATABUS is pulled down to 0. Thus this can be used as a register with 0 stored in it. The leafcell used is *zero*.

Description of the Terminals		
Terminal Name	Signal Type	Function
DATABUS	N-bit bidirectional	data bus port
ZERO	input	high to make DATABUS zero
GND	ground	connects to ground

Behavioral Description

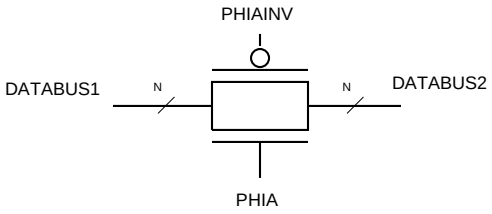
```
if (ZERO==1)
    for (i=0;i<=N-1;i++) DATABUS[i]=0;
else /*ZERO==0*/
    for (i=0;i<=N-1;i++) DATABUS[i]=HI_IMPEDANCE;
```

sdl File

```
(parent-cell zero)
(parameters N)
(layout-generator TimLager)
(terminal DATABUS (side LEFT RIGHT ))
(terminal GND (side BOT TOP ))
(terminal ZERO (side BOT TOP ))
(end-sdl)
```

Performance Results

Leafcell Delay Models				
Leafcell	From	To	Best Fit Linear Model	Root Mean Square Error
zero	ZERO	DATABUS	$t_{plh} = 0.0000 + 0.0000 * C_{load}$	0.0000
			$t_{phl} = 0.0558 + 0.9236 * C_{load}$	0.2134
			$t_{lh} = 0.0000 + 0.0000 * C_{load}$	0.0000
			$t_{hl} = 0.4721 + 1.4589 * C_{load}$	0.0511
Note: delays are in *ns*, C_{load} is in *pF*				



xfer_gate

Performance Results

Leafcell Delay Models				
Leafcell	From	To	Best Fit Linear Model	Root Mean Square Error
xfer_gate	PHIAINV	DATABUS2	$t_{plh} = 0.4160 + 3.8966 * C_{load}$	0.0494
			$t_{phl} = 0.3304 + 3.1484 * C_{load}$	0.0077
			$t_{lh} = 0.6701 + 14.1877 * C_{load}$	0.2641
			$t_{hl} = 0.4850 + 7.0195 * C_{load}$	0.2409
Note: delays are in ns, C_{load} is in pF				

27 xfer_gate

Synopsis

This implements N transfer gates in parallel. The leafcell used is *xfer_gate*.

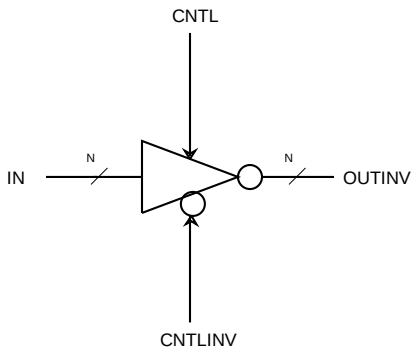
Description of the Terminals		
Terminal Name	Signal Type	Function
DATABUS1	N-bit bidirectional	data port 1
DATABUS2	N-bit bidirectional	data port 2
PHIA	input	clock
PHIAINV	input	inverted clock

Behavioral Description

```
if (PHIA==PHIAINV) ERROR();
else if (PHIA==1)
    if (DATABUS1 is being driven from outside)
        for (i=0;i<=N-1;i++) DATABUS2[i]=DATABUS1[i];
    else /*DATABUS2 is being driven from outside*/
        for (i=0;i<=N-1;i++) DATABUS1[i]=DATABUS2[i];
else /*PHIA==0*/
    for (i=0;i<=N-1;i++) DATABUS2[i]=DATABUS1[i]=HI_IMPEDANCE;
```

sdl File

```
(parent-cell xfer_gate)
(parameters N)
(layout-generator TimLager)
(terminal DATABUS1 (side LEFT RIGHT ))
(terminal DATABUS2 (side LEFT RIGHT ))
(terminal GND (side TOP BOT ))
(terminal PHIA (side TOP BOT ))
(terminal PHIAINV (side TOP BOT ))
(terminal Vdd (side TOP BOT ))
(end-sdl)
```



trist_inverter

Performance Results

Leafcell Delay Models				
Leafcell	From	To	Best Fit Linear Model	Root Mean Square Error
trist_inverter	IN	OUTINV	$t_{plh} = 0.5980 + 3.4825 * C_{load}$	0.0773
			$t_{phl} = 0.4618 + 2.5958 * C_{load}$	0.0319
			$t_{lh} = 0.8921 + 8.0492 * C_{load}$	0.2499
			$t_{hl} = 0.8187 + 5.4525 * C_{load}$	0.1901
trist_inverter	CNTL, CNTLINV	OUTINV	$t_{plh} = 0.3336 + 3.5330 * C_{load}$	0.0383
			$t_{phl} = 0.2836 + 2.6014 * C_{load}$	0.0222
			$t_{lh} = 0.9435 + 8.0454 * C_{load}$	0.2049
			$t_{hl} = 0.7577 + 5.4681 * C_{load}$	0.3872
Note: delays are in ns, C_{load} is in pF				

26 trist_inverter

Synopsis

This block implements a N-bit tristate inverter. the output is Hi-Z if the control signals CNTL and CNTLINV are not asserted to 1 and 0 respectively. It uses the leafcell *trist_inverter*.

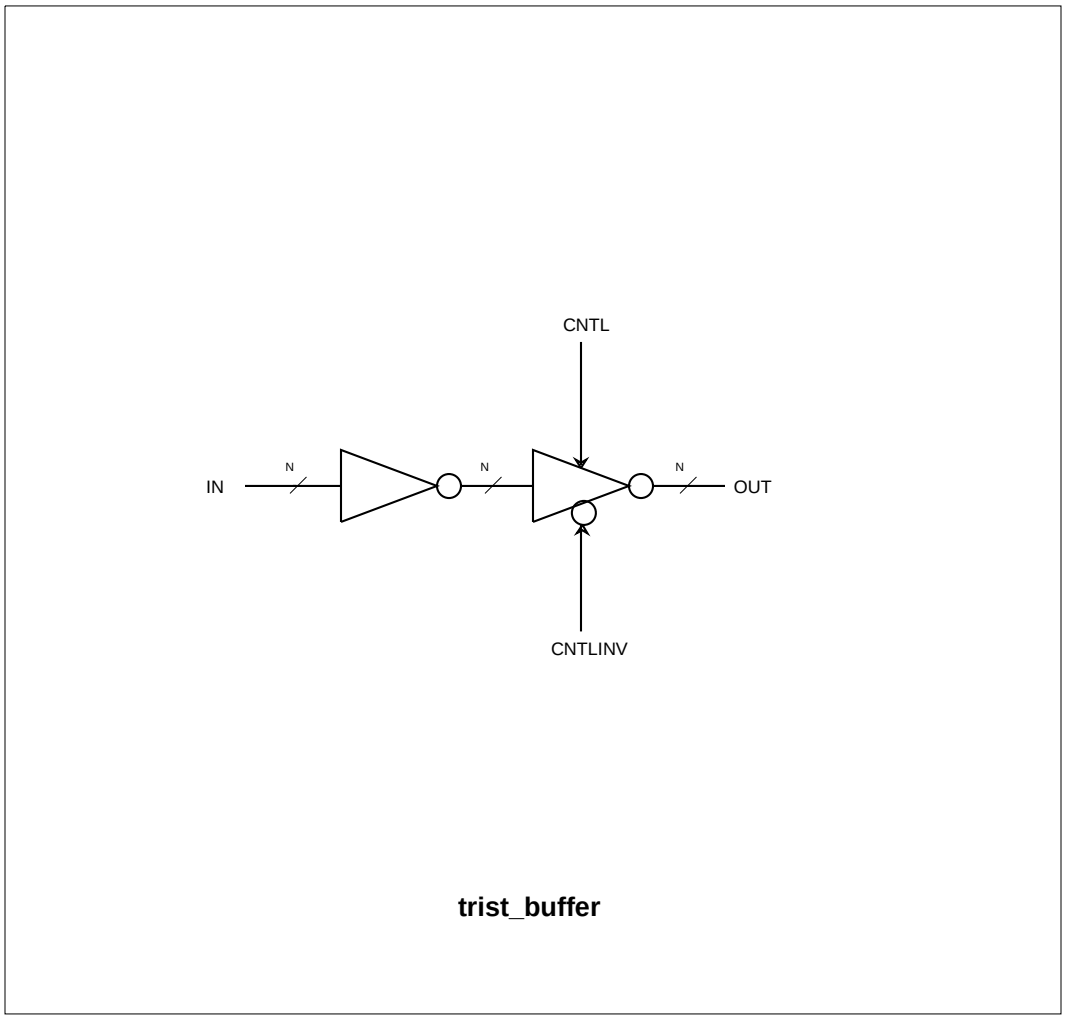
Description of the Terminals		
Terminal Name	Signal Type	Function
IN	N-bit input	data input port
OUTINV	N-bit output	data output port
CNTL	input	high to enable output
CNTLINV	input	low to enable output
Vdd	power	connects to power supply
GND	ground	connects to ground

Behavioral Description

```
if (CNTL==CNTLINV) ERROR();
else if (CNTL==1)
    for (i=0;i<=N-1;i++) OUT[i]=not(IN[i]);
else
    for (i=0;i<=N-1;i++) OUT[i]=HI_IMPEDANCE;
```

sdl File

```
(parent-cell trist_inverter)
(parameters N)
(layout-generator TimLager)
(terminal CNTL (side BOT TOP ))
(terminal CNTLINV (side BOT TOP ))
(terminal GND (side BOT TOP ))
(terminal IN (side LEFT RIGHT ))
(terminal OUTINV (side LEFT RIGHT ))
(terminal Vdd (side BOT TOP ))
(end-sdl)
```


CNTL
IN
N
N
N
OUT
CNTLINV
trist_buffer

Performance Results

Leafcell Delay Models				
Leafcell	From	To	Best Fit Linear Model	Root Mean Square Error
trist_buffer	IN	OUT	$t_{plh} = 0.6534 + 4.6503 * C_{load}$	0.2309
			$t_{phl} = 0.7428 + 2.3363 * C_{load}$	0.0552
			$t_{lh} = 0.9580 + 10.9458 * C_{load}$	0.1788
			$t_{hl} = 0.5590 + 5.0839 * C_{load}$	0.4600
trist_buffer	CNTL, CNTLINV	OUT	$t_{plh} = 0.3599 + 4.7293 * C_{load}$	0.0125
			$t_{phl} = 0.2495 + 2.3200 * C_{load}$	0.0329
			$t_{lh} = 0.9806 + 10.9214 * C_{load}$	0.1489
			$t_{hl} = 0.6198 + 5.0490 * C_{load}$	0.6229
Note: delays are in ns, C_{load} is in pF				

25 trist_buffer

Synopsis

This block implements a N-bit tristate buffer. the output is Hi-Z if the control signals CNTL and CNTLINV are not asserted to 1 and 0 respectively. It uses the leafcell *trist_buffer*.

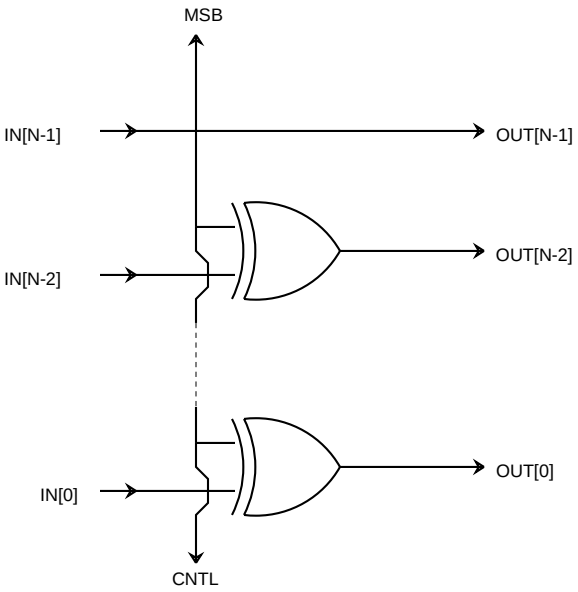
Description of the Terminals		
Terminal Name	Signal Type	Function
IN	N-bit input	data input port
OUT	N-bit output	data output port
CNTL	input	high to enable output
CNTLINV	input	low to enable output
Vdd	power	connects to power supply
GND	ground	connects to ground

Behavioral Description

```
if (CNTL==CNTLINV) ERROR();
else if (CNTL==1)
    for (i=0;i<=N-1;i++) OUT[i]=IN[i];
else
    for (i=0;i<=N-1;i++) OUT[i]=HI_IMPEDANCE;
```

sdl File

```
(parent-cell trist_buffer)
(parameters N)
(layout-generator TimLager)
(terminal GND (side BOT TOP ))
(terminal IN (side LEFT RIGHT ))
(terminal CNTL (side BOT TOP ))
(terminal CNTLINV (side BOT TOP ))
(terminal OUT (side RIGHT LEFT ))
(terminal Vdd (side BOT TOP ))
(end-sdl)
```



signmag

```
(terminal Vdd (side TOP BOT ))
(end-sdl)
```

Performance Results

Leafcell Delay Models				
Leafcell	From	To	Best Fit Linear Model	Root Mean Square Error
invpass	IN	OUT	$t_{plh} = 0.8182 + 5.4725 * C_{load}$	0.0299
			$t_{phl} = 0.4382 + 2.3188 * C_{load}$	0.0174
			$t_{lh} = 1.6205 + 12.8434 * C_{load}$	0.1337
			$t_{hl} = 1.0290 + 4.8059 * C_{load}$	0.1562
	CNTL	OUT	$t_{plh} = 0.8221 + 5.4809 * C_{load}$	0.0216
			$t_{phl} = 0.4775 + 2.3244 * C_{load}$	0.0301
			$t_{lh} = 1.3456 + 12.8262 * C_{load}$	0.1241
			$t_{hl} = 0.9835 + 4.7990 * C_{load}$	0.0726
Note: delays are in ns, C_{load} is in pF				

24 signmag

Synopsis

This block converts a one's complement number into a sign magnitude number. The most significant bit, MSB, passes through the block unchanged and is used as a control signal (CNIL) for less significant bits. If CNIL is 0, then the input is positive and OUT equals IN; however, if CNIL is 1, the input is negative and the N-1 least significant bits are inverted for output. The block uses the leafcell *signmag_msb* in the MSB position and the leafcell *invpass* in the remaining bit positions.

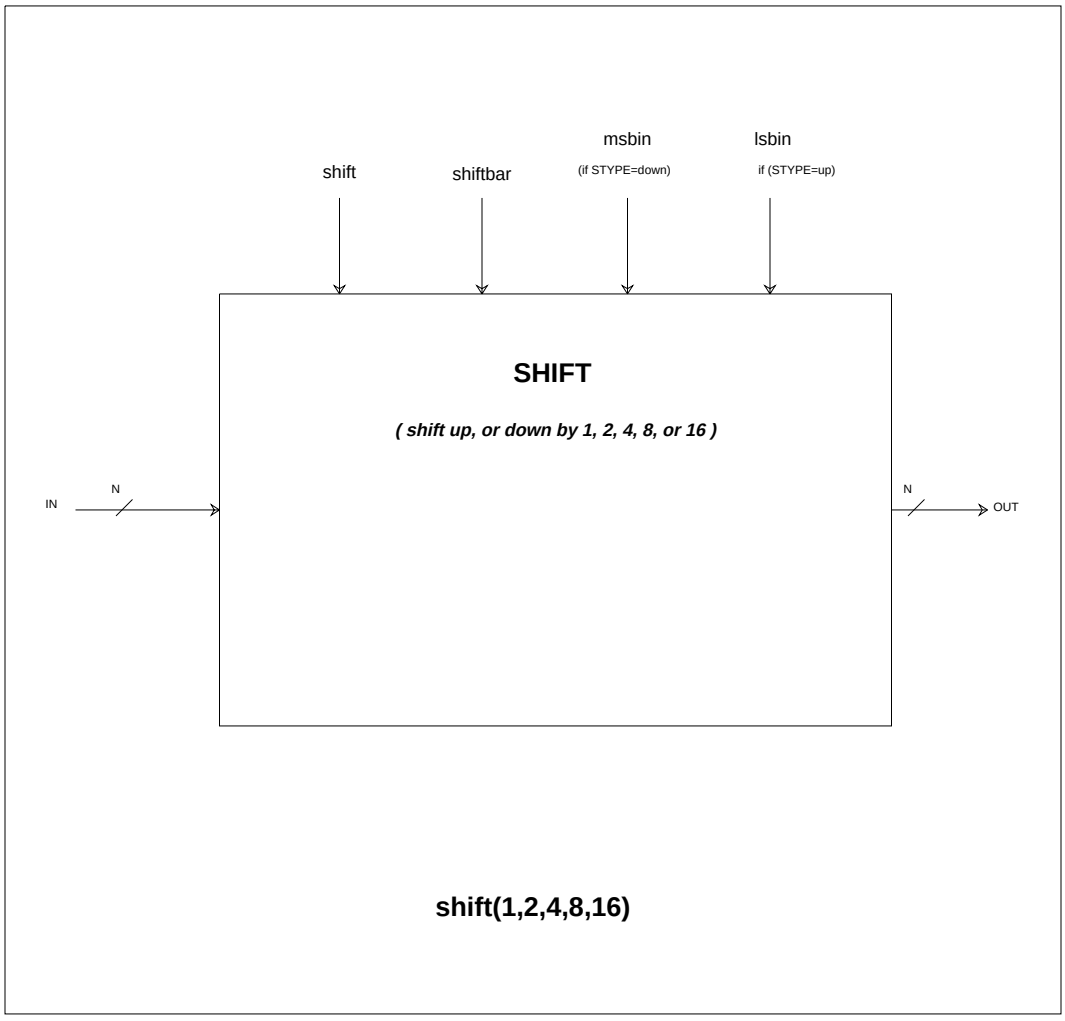
Description of the Terminals		
Terminal Name	Signal Type	Function
IN	N-bit input	two's complement input
OUT	N-bit output	sign-magnitude output
MSB	output	most significant bit
CNIL	output	most significant bit
Vdd	power	connects to power supply
GND	ground	connects to ground

Behavioral Description

```
OUT[N-1]=IN[N-1];
for (i=0;i<=N-2;i++) {
    if (IN[N-1]==0) {
        /* positive number */
        OUT[i] = IN[i]
    } else {
        /* negative number */
        OUT[i] = !(IN[i])
    }
}
```

sdl File

```
(parent-cell signmag)
(parameters N)
(layout-generator TimLager)
(terminal SEL1 (side TOP BOT ))
(terminal SEL2 (side TOP BOT ))
(terminal GND (side TOP BOT ))
(terminal IN1 (side RIGHT LEFT ))
(terminal IN2 (side RIGHT LEFT ))
(terminal OUT (side RIGHT ))
(terminal OUTINV (side LEFT ))
```


msbin
lsbin
shift
shiftbar
(if STYPE=down)
if (STYPE=up)
SHIFT
(shift up, or down by 1, 2, 4, 8, or 16)
IN
N
N
OUT
shift(1,2,4,8,16)

Leafcell Propagation Delays							
IN	OUT	Conditions	Load (pF)	Tplh (ns)	Tphl (ns)	Tlh (ns)	Thl (ns)
shift	IN	OUT	t_{plh} =1.1448 +3.5075 * C_{load}	0.0780			
			t_{phl} =1.3786 +2.3965 * C_{load}	0.1042			
			t_{lh} =0.5779 +7.9728 * C_{load}	0.2741			
			t_{hl} =0.9732 +4.3920 * C_{load}	0.5176			
shift	PREV	OUT	t_{plh} =0.9370 +3.4980 * C_{load}	0.0411			
			t_{phl} =1.0791 +2.4009 * C_{load}	0.0990			
			t_{lh} =0.5517 +7.9715 * C_{load}	0.1326			
			t_{hl} =0.9742 +4.3837 * C_{load}	0.2753			
shift	SHIFT	OUT	t_{plh} =1.1976 +3.4974 * C_{load}	0.0698			
			t_{phl} =1.1709 +2.3991 * C_{load}	0.1157			
			t_{lh} =0.5967 +7.9593 * C_{load}	0.3317			
			t_{hl} =0.9941 +4.3741 * C_{load}	0.3272			
shift	SHIFT	OUT	t_{plh} =0.9416 +3.5008 * C_{load}	0.0455			
			t_{phl} =1.0796 +2.4053 * C_{load}	0.1000			
			t_{lh} =0.5491 +7.9716 * C_{load}	0.1397			
			t_{hl} =0.9929 +4.3776 * C_{load}	0.2870			
Note: delays are in *ns*, C_{load} is in *pF*							

```
(terminal SHIFTBAR (side TOP BOT)(TERMTYPE CONTROL)
(DIRECTION INPUT))
(terminal Vdd (side TOP BOT)(TERMTYPE SUPPLY))
(terminal GND (side TOP BOT)(TERMTYPE GROUND))
(terminal MSBIN (side TOP BOT)(TERMTYPE CONTROL)(DIRECTION INPUT))
(terminal LSBIN (side TOP BOT)(TERMTYPE CONTROL)(DIRECTION INPUT))
(net msbin (CONDITIONAL (if (eq STYPE "down") 1 0))((parent MSBIN)))
(net lsbin (CONDITIONAL (if (eq STYPE "up") 1 0))((parent LSBIN)))
(end-sdl)
```

23 shift

Synopsis

This block implements a parameterized shifter. The direction of shift is specified by the parameter STYPE which is set to one of the the following string values: "up" (LSBIN specifies what is shifted into least significant bit), "down" (MSBIN specifies what is shifted into most significant bit), or "sxdown" (sign extends most significant bit). The amount to be shifted by is past as the string parameter SBY. Current valid SBY values are 1, 2, 4, 8, and 16. This block uses the leafcells *down#* *down#msb* *down#lsb* *up#* *up#msb* *up#lsb* and *sxdown#msb* where # is 1, 2, 4, 8, or 16.

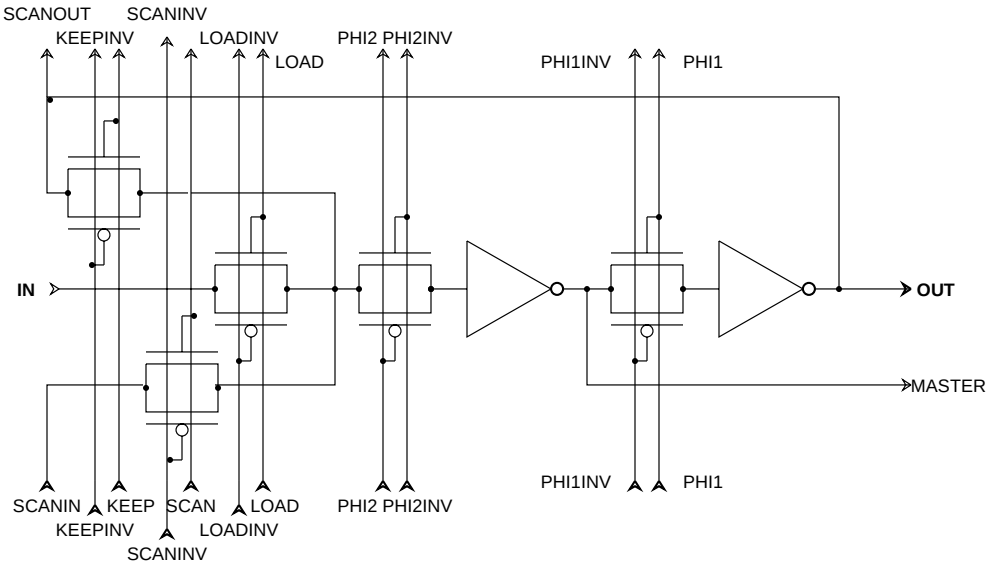
Description of the Terminals		
Terminal Name	Signal Type	Function
IN	N-bit input	input data
OUT	N-bit output	output data
SHIFT	input	control signal
SHIFTBAR	input	control signal
MSBIN	Cond. input	control signal (if STYPE="down")
LSBIN	Cond. input	control signal (if STYPE="up")
Vdd	power	connects to power supply
GND	ground	connects to ground

Thor Model

```
if (SHIFT==ONE) && (SHIFTBAR==ZERO);
  if (!STYPE!=="down") fshftr(OUT,!N!-1,0,!SBY!,MSBIN);
  else
    if (!STYPE!=="up") fshftr(OUT,!N!-1,0,!SBY!,LSBIN);
    else
      if (!STYPE!=="sxdown") fshftr(OUT,!N!-1,0,!SBY!,IN[!N1-1]);
else
  fsetword(OUT,!N!-1,0,UNDEF);
EXITMOD(0);
```

sdl File

```
(parent-cell shift)
(parameters N SBY STYPE (MODULE_TYPE "SHIFTER"))
(layout-generator TimLager)
(terminal IN  (side LEFT)(TERMTYPE DATA_SIGNAL)(DIRECTION INPUT))
(terminal OUT (side RIGHT)(TERMTYPE DATA_SIGNAL)(DIRECTION OUTPUT))
(terminal SHIFT (side TOP BOT)(TERMTYPE CONTROL)
(DIRECTION INPUT))
```


SCANOUT
SCANINV
KEEPINV
LOADINV
PHI2 PHI2INV
LOAD
PHI1INV
PHI1
IN
OUT
MASTER
SCANIN
KEEP
SCAN
LOAD
KEEPINV
LOADINV
SCANINV
PHI2 PHI2INV
PHI1INV
PHI1

scanregmx

Performance Results

Leafcell Delay Models				
Leafcell	From	To	Best Fit Linear Model	Root Mean Square Error
scanregmx	LOAD	OUT	$t_{plh} = 1.6462 + 1.7940 * C_{load}$	0.0089
			$t_{phl} = 1.7298 + 2.0563 * C_{load}$	0.0509
			$t_{lh} = 0.6650 + 4.0732 * C_{load}$	0.1992
			$t_{hl} = 0.7426 + 3.7853 * C_{load}$	0.1528
scanregmx	PHI1	OUT	$t_{plh} = 0.7127 + 1.8096 * C_{load}$	0.0342
			$t_{phl} = 0.9313 + 2.0378 * C_{load}$	0.0218
			$t_{lh} = 0.7179 + 4.0508 * C_{load}$	0.1930
			$t_{hl} = 0.7282 + 3.7872 * C_{load}$	0.1821
scanregmx	PHI2	OUT	$t_{plh} = 0.4253 + 1.7192 * C_{load}$	0.0297
			$t_{phl} = 1.2170 + 2.0590 * C_{load}$	0.0612
			$t_{lh} = 1.1192 + 3.8959 * C_{load}$	0.3067
			$t_{hl} = 0.7277 + 3.7924 * C_{load}$	0.1429
scanregmx	LOAD	OUT	$t_{plh} = 2.0068 + 1.8007 * C_{load}$	0.0384
			$t_{phl} = 2.1119 + 2.0762 * C_{load}$	0.0860
			$t_{lh} = 0.7908 + 4.0261 * C_{load}$	0.2001
			$t_{hl} = 0.8975 + 3.7414 * C_{load}$	0.1683
scanregmx	PHI1	OUT	$t_{plh} = 0.7073 + 1.8124 * C_{load}$	0.0411
			$t_{phl} = 0.8936 + 2.0550 * C_{load}$	0.0423
			$t_{lh} = 0.7246 + 4.0491 * C_{load}$	0.1836
			$t_{hl} = 0.7462 + 3.7866 * C_{load}$	0.1577
scanregmx	PHI2	OUT	$t_{plh} = 0.6805 + 1.7580 * C_{load}$	0.0407
			$t_{phl} = 1.5890 + 2.0692 * C_{load}$	0.0674
			$t_{lh} = 0.9871 + 3.9614 * C_{load}$	0.1629
			$t_{hl} = 0.8801 + 3.7487 * C_{load}$	0.1648
Note: delays are in ns, C_{load} is in pF				

```
(terminal OUT      (side RIGHT LEFT))
(terminal MASTER   (side RIGHT LEFT))
(terminal SCANIN   (side BOTTOM))
(terminal SCANOUT  (side TOP))
;
(terminal LOAD     (side TOP BOTTOM))
(terminal LOADINV  (side TOP BOTTOM))
(terminal SCAN     (side TOP BOTTOM))
(terminal SCANINV  (side TOP BOTTOM))
(terminal KEEP     (side TOP BOTTOM))
(terminal KEEPINV  (side TOP BOTTOM))
(terminal PHI2     (side TOP BOTTOM))
(terminal PHI2INV  (side TOP BOTTOM))
(terminal PHI1     (side TOP BOTTOM))
(terminal PHI1INV  (side TOP BOTTOM))
;
(terminal Vdd      (side TOP BOTTOM))
(terminal GND      (side TOP BOTTOM))
;
(end-sdl)
```

```
   }
   else if (((LOAD==1)||(LOADINV==0)) && ((KEEP==1)||(KEEPINV==0))) {
      ERRMESS("Both LOAD and KEEP are active.\n");
      EXITMOD(0);
   }

   if ((PHI2==1) && (PHI2INV==0)) {
      /* LOAD operation */
      if ((LOAD==1) && (LOADINV==0))
         fcopy(REGm, !N!-1, 0, IN, !N!-1, 0);

      /* load scan data from input node SCAN */
      if ((SCAN== 1) && (SCANINV == 0))
         {
         fcopy(REGm, !N!-1, 0, REGs, !N!-1, 0);
         fshftl(REGm, !N!-1, 0, 1, SCANIN);
         }

      /* keep the old register contents (redundant here, but ...)  */
      if ((KEEP==1) && (KEEPINV==0))
         fcopy(REGm, !N!-1, 0, REGs, !N!-1, 0);
   }

   /* update slave register from master with PHI1 */
   if ((PHI1 == 1) && (PHI1INV == 0))
      fcopy(REGs, !N!-1, 0,REGm, !N!-1, 0);

   /* Handle the outputs. */
   fcopy(OUT, !N!-1, 0, REGs, !N!-1, 0);
   fcopy(MASTER, !N!-1, 0, REGm, !N!-1, 0);
   SCANOUT = REGs[!N!-1];

EXITMOD(0);
}
```

sdl File

```
(parent-cell scanregmx)
(layout-generator TimLager)
(structure-processor dpp)
;
(parameters N)
;
(terminal IN       (side RIGHT LEFT))
```

22 scanregmx

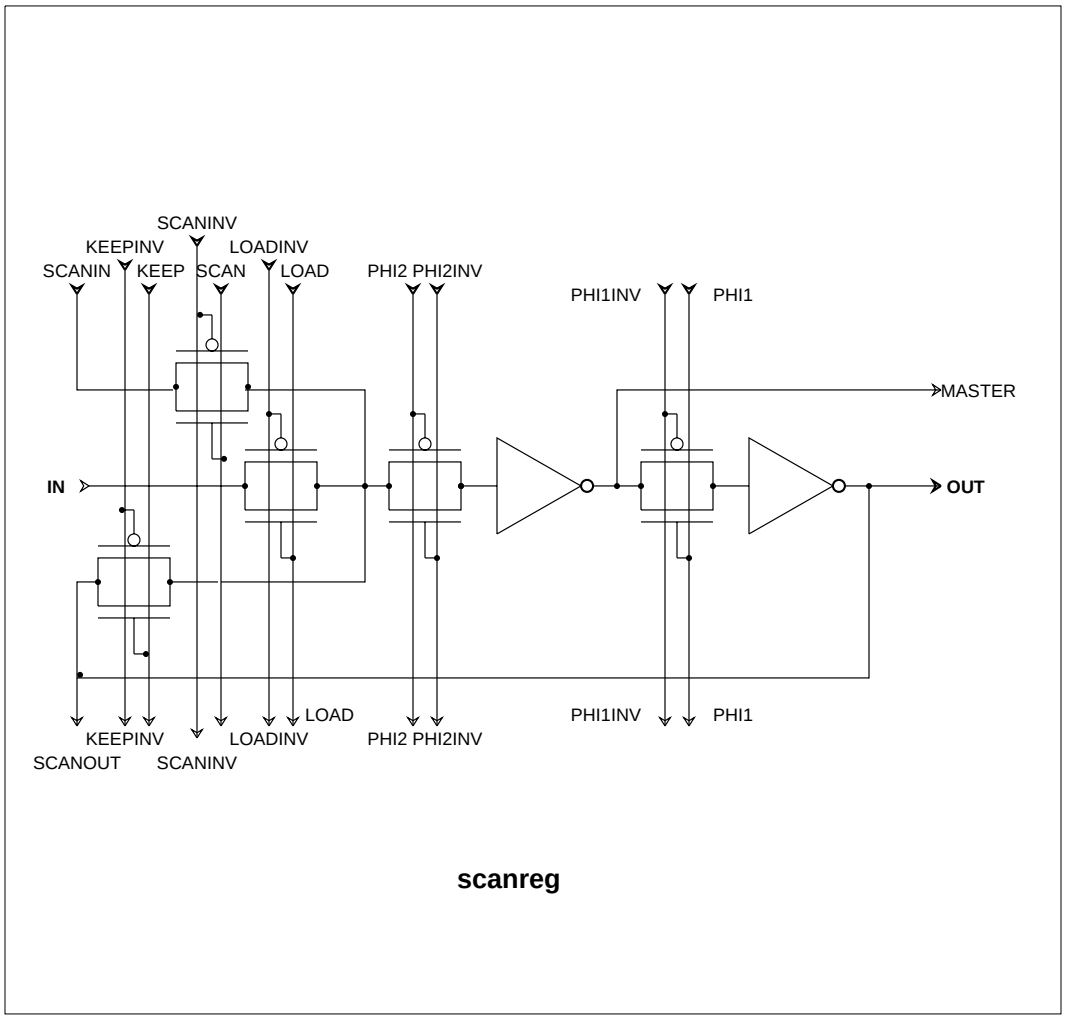
Synopsis

This is a N-bit single-ported scan path register. The input nodes of this register are either IN during the normal mode of operation (latched with LOAD and LOADINV) or SCANIN in the testmode (latched with SCAN and SCANINV). To achieve a quasi static behavior the register can feed back the output if the KEEP and KEEPINV control signals are used. It is recommended to generate the KEEP as well as the KEEPINV signal using the LOAD and SHIFT signals so that the register can keep the value if both of these signals are not valid. During normal operation IN as well as all the control signals have to be valid *before* the falling edge of PHI2 (rising edge of PHI2INV). The data at the outputs OUT and SCANOUT are valid after the rising edge of PHI1 (falling edge of PHI1INV). The scanpath proceeds from the LSB to the MSB. This block is functionally equivalent to *scanreg* block except that the latter has the scanpath going from the MSB to LSB. The leafcell used is *scanregmx*.

Description of the Terminals		
Terminal Name	Signal Type	Function
IN	N-bit input	data input port
OUT	N-bit output	positive logic data output port
SCANIN	input	scanpath serial input
SCANOUT	output	scanpath serial output
LOAD	input	high to load the register
LOADINV	input	low to load the register
SCAN	input	high during testmode
SCANINV	input	low during testmode
KEEP	input	high if register value is fed back
KEEPINV	input	low if register value is fed back
PHI2	input	master clock
PHI2INV	input	inverted master clock
PHI1	input	slave clock
PHI1INV	input	inverted slave clock
Vdd	power	connects to power supply
GND	ground	connects to ground

Behavioral Description

```
/* Check error conditions  */
   if (((SCAN==1)||(SCANINV==0)) && ((LOAD==1)||(LOADINV==0))) {
      ERRMESS("Both LOAD and SCAN are active.\n");
      EXITMOD(0);
   }
   else if (((SCAN==1)||(SCANINV==0)) && ((KEEP==1)||(KEEPINV==0))) {
      ERRMESS("Both SCAN and KEEP are active.\n");
      EXITMOD(0);
```


SCANINV
KEEPINV
LOADINV
SCANIN
KEEP
SCAN
LOAD
PHI2 PHI2INV
PHI1INV
PHI1
MASTER
IN
OUT
LOAD
KEEPINV
LOADINV
PHI2 PHI2INV
PHI1INV
PHI1
SCANOUT
SCANINV

scanreg

Performance Results

Leafcell Delay Models				
Leafcell	From	To	Best Fit Linear Model	Root Mean Square Error
scanreg	LOAD	OUT	$t_{plh} = 1.6701 + 2.2707 * C_{load}$	0.0125
			$t_{phl} = 1.6772 + 2.0491 * C_{load}$	0.0460
			$t_{lh} = 0.6676 + 5.2121 * C_{load}$	0.1610
			$t_{hl} = 0.7405 + 3.7800 * C_{load}$	0.1797
scanreg	PHI1	OUT	$t_{plh} = 0.7352 + 2.2779 * C_{load}$	0.0073
			$t_{phl} = 0.8431 + 2.0506 * C_{load}$	0.0402
			$t_{lh} = 0.7506 + 5.1751 * C_{load}$	0.2246
			$t_{hl} = 0.7116 + 3.7912 * C_{load}$	0.1680
scanreg	PHI2	OUT	$t_{plh} = 0.5192 + 2.1886 * C_{load}$	0.0323
			$t_{phl} = 1.1724 + 2.0562 * C_{load}$	0.0542
			$t_{lh} = 0.9791 + 5.0862 * C_{load}$	0.2183
			$t_{hl} = 0.7317 + 3.7862 * C_{load}$	0.1919
scanreg	LOAD	OUT	$t_{plh} = 2.0460 + 2.2746 * C_{load}$	0.0216
			$t_{phl} = 2.0486 + 2.0745 * C_{load}$	0.0955
			$t_{lh} = 0.7794 + 5.1713 * C_{load}$	0.1857
			$t_{hl} = 0.8829 + 3.7386 * C_{load}$	0.1748
scanreg	PHI1	OUT	$t_{plh} = 0.7219 + 2.2835 * C_{load}$	0.0200
			$t_{phl} = 0.8175 + 2.0634 * C_{load}$	0.0630
			$t_{lh} = 0.7626 + 5.1706 * C_{load}$	0.2218
			$t_{hl} = 0.7180 + 3.7995 * C_{load}$	0.1431
scanreg	PHI2	OUT	$t_{plh} = 0.8028 + 2.2119 * C_{load}$	0.0288
			$t_{phl} = 1.5314 + 2.0718 * C_{load}$	0.0839
			$t_{lh} = 0.9934 + 5.0820 * C_{load}$	0.2728
			$t_{hl} = 0.8619 + 3.7519 * C_{load}$	0.1709
Note: delays are in ns, C_{load} is in pF				

```
(terminal SCANOUT  (side BOTTOM))
;
(terminal LOAD     (side TOP BOTTOM))
(terminal LOADINV  (side TOP BOTTOM))
(terminal SCAN     (side TOP BOTTOM))
(terminal SCANINV  (side TOP BOTTOM))
(terminal KEEP     (side TOP BOTTOM))
(terminal KEEPINV  (side TOP BOTTOM))
(terminal PHI2     (side TOP BOTTOM))
(terminal PHI2INV  (side TOP BOTTOM))
(terminal PHI1     (side TOP BOTTOM))
(terminal PHI1INV  (side TOP BOTTOM))
;
(terminal Vdd      (side TOP BOTTOM))
(terminal GND      (side TOP BOTTOM))
;
(end-sdl)
```

```
   }
   else if (((LOAD==1)||(LOADINV==0)) && ((KEEP==1)||(KEEPINV==0))) {
      ERRMESS("Both LOAD and KEEP are active.\n");
      EXITMOD(0);
   }

   if ((PHI2==1) && (PHI2INV==0)) {
      /* LOAD operation */
      if ((LOAD==1) && (LOADINV==0))
         fcopy(REGm, !N!-1, 0, IN, !N!-1, 0);

      /* load scan data from input node SCAN */
      if ((SCAN== 1) && (SCANINV == 0))
         {
         fcopy(REGm, !N!-1, 0, REGs, !N!-1, 0);
         fshftr(REGm, !N!-1, 0, 1, SCANIN);
         }

      /* keep the old register contents (redundant here, but ...)  */
      if ((KEEP==1) && (KEEPINV==0))
         fcopy(REGm, !N!-1, 0, REGs, !N!-1, 0);
   }

   /* update slave register from master with PHI1 */
   if ((PHI1 == 1) && (PHI1INV == 0))
      fcopy(REGs, !N!-1, 0,REGm, !N!-1, 0);

   /* Handle the outputs. */
   fcopy(OUT, !N!-1, 0, REGs, !N!-1, 0);
   fcopy(MASTER, !N!-1, 0, REGm, !N!-1, 0);
   SCANOUT = REGs[!N!-1];
```

sdl File

```
(parent-cell scanreg)
(layout-generator TimLager)
(structure-processor dpp)
;
(parameters N)
;
(terminal IN       (side RIGHT LEFT))
(terminal OUT      (side RIGHT LEFT))
(terminal MASTER   (side RIGHT LEFT))
(terminal SCANIN   (side TOP))
```

21 scanreg

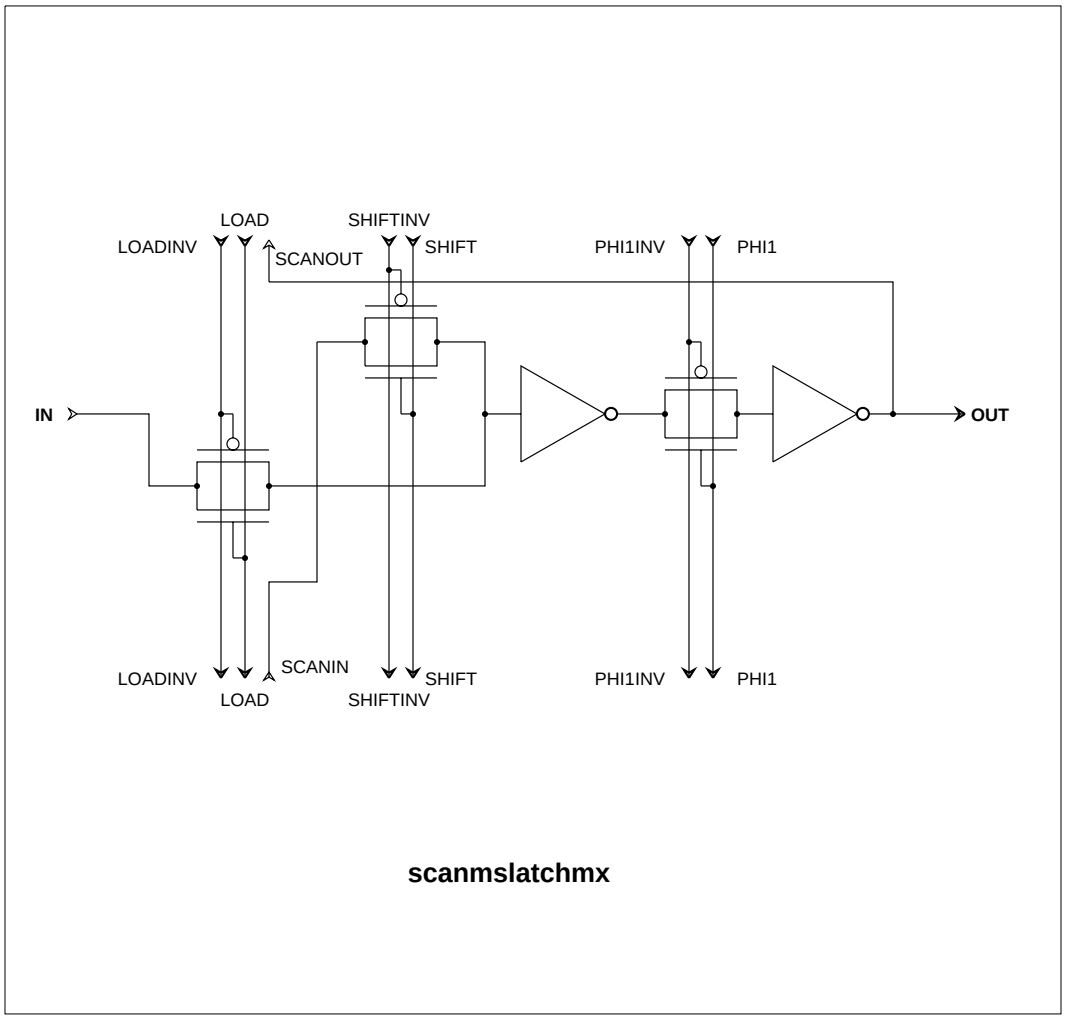
Synopsis

This is a N-bit single-ported scan path register. The input nodes of this register are either IN during the normal mode of operation (latched with LOAD and LOADINV) or SCANIN in the testmode (latched with SCAN and SCANINV). To achieve a quasi static behavior the register can feed back the output if the KEEP and KEEPINV control signals are used. It is recommended to generate the KEEP as well as the KEEPINV signal using the LOAD and SHIFT signals so that the register can keep the value if both of these signals are not valid. During normal operation IN as well as all the control signals have to be valid *before* the falling edge of PHI2 (rising edge of PHI2INV). The data at the outputs OUT and SCANOUT are valid after the rising edge of PHI1 (falling edge of PHI1INV). The scanpath proceeds from the MSB to the LSB. This block is functionally equivalent to *scanregmr* block except that the latter has the scanpath going from the LSB to MSB. The leafcell used is *scanreg*.

Description of the Terminals		
Terminal Name	Signal Type	Function
IN	N-bit input	data input port
OUT	N-bit output	positive logic data output port
SCANIN	input	scanpath serial input
SCANOUT	output	scanpath serial output
LOAD	input	high to load the register
LOADINV	input	low to load the register
SCAN	input	high during testmode
SCANINV	input	low during testmode
KEEP	input	high if register value is fed back
KEEPINV	input	low if register value is fed back
PHI2	input	master clock
PHI2INV	input	inverted master clock
PHI1	input	slave clock
PHI1INV	input	inverted slave clock
Vdd	power	connects to power supply
GND	ground	connects to ground

Behavioral Description

```
/* Check error conditions  */
   if (((SCAN==1)||(SCANINV==0)) && ((LOAD==1)||(LOADINV==0))) {
      ERRMESS("Both LOAD and SCAN are active.\n");
      EXITMOD(0);
   }
   else if (((SCAN==1)||(SCANINV==0)) && ((KEEP==1)||(KEEPINV==0))) {
      ERRMESS("Both SCAN and KEEP are active.\n");
      EXITMOD(0);
```



scanmslatchmx

Performance Results

Leafcell Delay Models				
Leafcell	From	To	Best Fit Linear Model	Root Mean Square Error
scanmslatchmx	SHIFT	OUT	$t_{plh} = 1.3385 + 1.2112 * C_{load}$	0.0280
			$t_{phl} = 1.3024 + 1.0806 * C_{load}$	0.0885
			$t_{lh} = 0.4196 + 2.7590 * C_{load}$	0.0819
			$t_{hl} = 0.4922 + 1.8979 * C_{load}$	0.0446
scanmslatchmx	PHI1	OUT	$t_{plh} = 0.6606 + 1.2067 * C_{load}$	0.0304
			$t_{phl} = 0.7878 + 1.0778 * C_{load}$	0.0816
			$t_{lh} = 0.4249 + 2.7562 * C_{load}$	0.0886
			$t_{hl} = 0.5286 + 1.8819 * C_{load}$	0.0558
Note: delays are in ns, C_{load} is in pF				

```
       * if not set REG to DATAIN.
       */
      if ((SHIFT == 1) && (SHIFTINV == 0))
         ERRMESS("Both Scanpath and Datapath are active.\n");
         else
         fcopy(REGm, !N!-1, 0, IN, !N!-1, 0);
      }

/*
 * PHI1 always takes the contents of the master
 * register to update the slave
 */
   if ((PHI1 == 1) && (PHI1INV == 0))
      fcopy(REGs, !N!-1, 0,REGm, !N!-1, 0);

/* Handle the outputs. */
   fcopy(OUT, !N!-1, 0,REGs, !N!-1, 0);
   SCANOUT = REGs[0];
```

sdl File

```
(parent-cell scanmslatchmx)
(parameters N (FEEDTHRUS 0)(BITHEIGHT 0)(MODULE_TYPE "SCANMSLATCHMX"))
(layout-generator TimLager)
(structure-processor dpp)
;
(terminal IN       (side RIGHT LEFT))
(terminal OUT      (side RIGHT LEFT))
(terminal SCANIN   (side BOTTOM))
(terminal SCANOUT  (side TOP))
;
(terminal LOAD     (side TOP BOTTOM))
(terminal LOADINV  (side TOP BOTTOM))
(terminal SHIFT    (side TOP BOTTOM))
(terminal SHIFTINV (side TOP BOTTOM))
(terminal PHI1     (side TOP BOTTOM))
(terminal PHI1INV  (side TOP BOTTOM))
;
(terminal Vdd      (side TOP BOTTOM))
(terminal GND      (side TOP BOTTOM))
;
(end-sdl)
```

20 scanmslatchmx

Synopsis

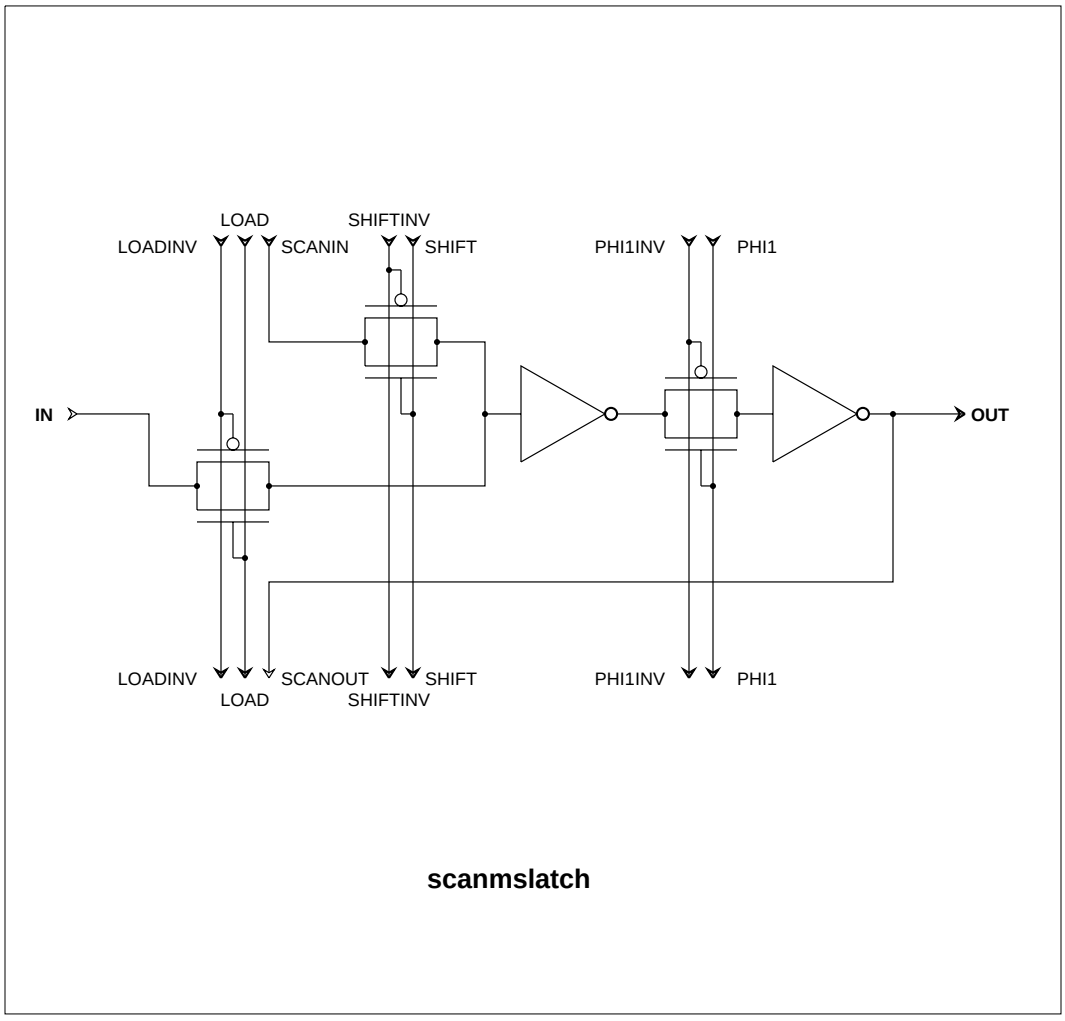
This is a N-bit single-ported dynamic scan path register. Due to the dynamic character it is meant to be used as a pipelineregister. However, if the output is fed back to the input via a multiplexor it can implement a static register with conditional load. The input nodes of this register are either IN during the normal mode of operation (latched with LOAD and LOADINV) or SCANIN in the testmode (latched with SCAN and SCANINV).
During normal operation IN has to be valid before the falling edge of LOAD (rising edge of LOADINV) while during the testmode it's value is insignificant (LOAD=0, LOADINV =1). The slave clocks PHI1 and PHI1INV are operating during both modes of operation so that only 6 clocks have to be supplied to the registercell. The data at the outputs OUT and SCANOUT are valid after the rising edge of PHI1 (falling edge of PHI1INV). The scanpath proceeds fromthe LSB to the MSB. This block is functionally equivalent to *scanmslatch* block except that the latter has the scanpath going fromthe MSB to LSB. The leafcell used is *scanmslatchmx*.

Description of the Terminals		
Terminal Name	Signal Type	Function
IN	N-bit input	data input port
OUT	N-bit output	positive logic data output port
SCANIN	input	scanpath serial input
SCANOUT	output	scanpath serial output
LOAD	input	high to load the register
LOADINV	input	low to load the register
SHIFT	input	test clock (master)
SHIFTINV	input	inverted test clock (master)
PHI1	input	slave clock
PHI1INV	input	inverted slave clock
Vdd	power	connects to power supply
GND	ground	connects to ground

Behavioral Description

```
/* load scan data from input node SCAN */
   if ((SHIFT== 1) && (SHIFTINV == 0)) {
      fcopy(REGm, !N!-1, 0, REGs, !N!-1, 0);
      fshftr(REGm, !N!-1, 0, 1, SCANIN);
      }

/* Data is input through the datapath. */
   if ((LOAD == 1) && (LOADINV == 0)) {
      /*
       * Check to see if Scan path is active,
```



scanmslatch

Performance Results

Leafcell Delay Models				
Leafcell	From	To	Best Fit Linear Model	Root Mean Square Error
scanmslatch	SHIFT	OUT	$t_{plh} = 1.3405 + 1.0800 * C_{load}$	0.0410
			$t_{phl} = 1.3383 + 1.0822 * C_{load}$	0.0845
			$t_{lh} = 0.4273 + 2.4441 * C_{load}$	0.0765
			$t_{hl} = 0.5142 + 1.8934 * C_{load}$	0.0416
scanmslatch	PHI1	OUT	$t_{plh} = 0.6712 + 1.0745 * C_{load}$	0.0272
			$t_{phl} = 0.8431 + 1.0750 * C_{load}$	0.0749
			$t_{lh} = 0.4381 + 2.4385 * C_{load}$	0.0893
			$t_{hl} = 0.5465 + 1.8790 * C_{load}$	0.0551
Note: delays are in ns, C_{load} is in pF				

```
/* load scan data from input node SCAN */
   if ((SHIFT== 1) && (SHIFTINV == 0))
      {
      fcopy(REGm, !N!-1, 0, REGs, !N!-1, 0);
      fshftl(REGm, !N!-1, 0, 1, SCANIN);
      }

/*
 * PHI1 always takes the contents of the master
 * register to update the slave
 */
   if ((PHI1 == 1) && (PHI1INV == 0))
      fcopy(REGs, !N!-1, 0,REGm, !N!-1, 0);

/* Handle the outputs. */
   fcopy(OUT, !N!-1, 0, REGs, !N!-1, 0);
   SCANOUT = REGs[!N!-1];
```

sdl File

```
(parent-cell scanmslatch)
(parameters N (FEEDTHRUS 0) (BITHEIGHT 0)(MODULE_TYPE "SCANMSLATCH"))
(layout-generator TimLager)
(structure-processor dpp)
;
;
(terminal IN       (side RIGHT LEFT))
(terminal OUT      (side RIGHT LEFT))
(terminal SCANIN   (side TOP))
(terminal SCANOUT  (side BOTTOM))
;
(terminal LOAD     (side TOP BOTTOM))
(terminal LOADINV  (side TOP BOTTOM))
(terminal SHIFT    (side TOP BOTTOM))
(terminal SHIFTINV (side TOP BOTTOM))
(terminal PHI1     (side TOP BOTTOM))
(terminal PHI1INV  (side TOP BOTTOM))
;
(terminal Vdd      (side TOP BOTTOM))
(terminal GND      (side TOP BOTTOM))
;
(end-sdl)
```

19 scanmslatch

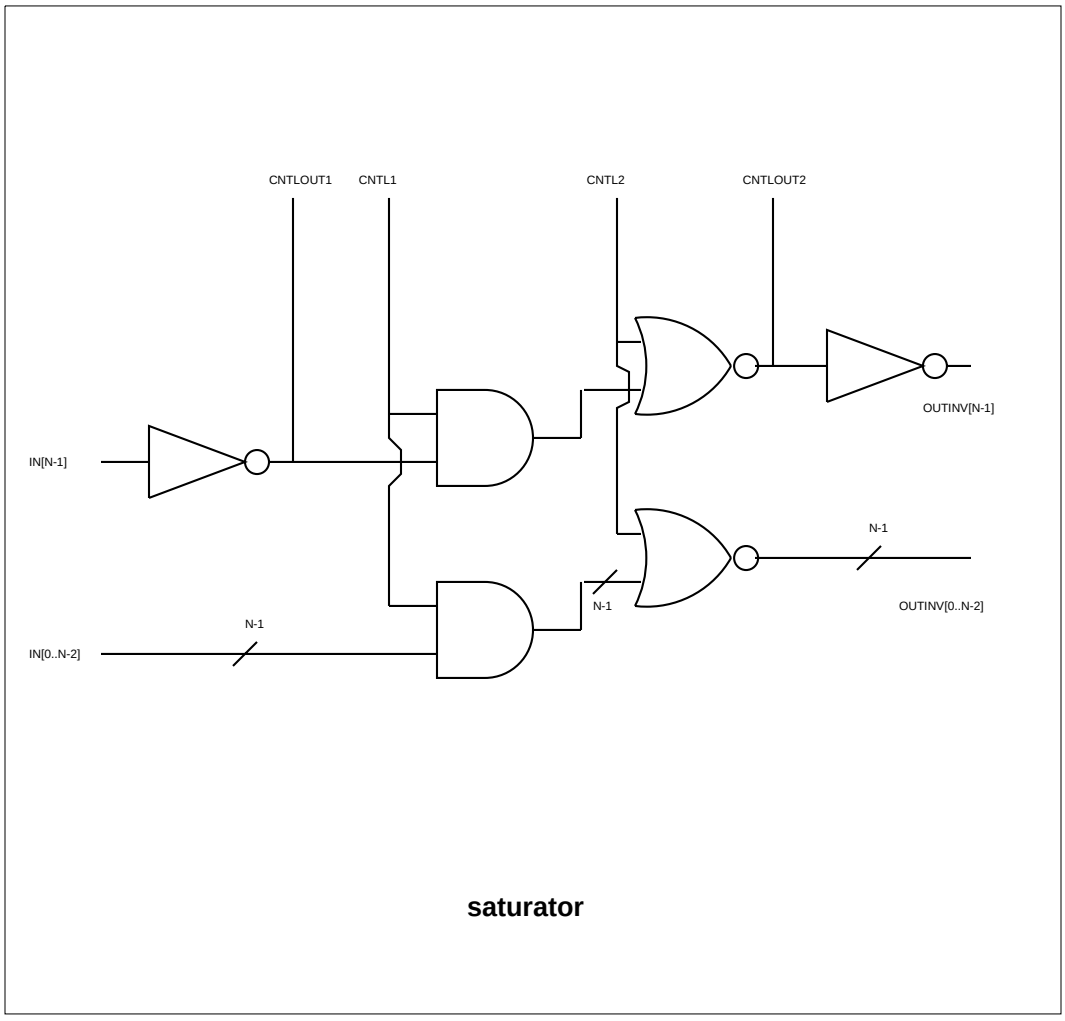
Synopsis

This is a N-bit single-ported dynamic scan path register. Due to the dynamic character it is meant to be used as a pipeline register. However, if the output is fed back to the input via a multiplexor it can implement a static register with conditional load. The input nodes of this register are either IN during the normal mode of operation (latched with LOAD and LOADINV) or SCANIN in the testmode (latched with SCAN and SCANINV).
During normal operation IN has to be valid before the falling edge of LOAD (rising edge of LOADINV) while during the testmode it's value is insignificant (LOAD=0, LOADINV=1).
The slave clocks PHI1 and PHI1INV are operating during both modes of operation so that only 6 clocks have to be supplied. The data at the outputs OUT and SCANOUT are valid after the rising edge of PHI1 (falling edge of PHI1INV). The scanpath proceeds from the MSB to the LSB. This block is functionally equivalent to *scanmslatchmr* block except that the latter has the scanpath going from the LSB to MSB. The leafcell used is *scanmslatch*.

Description of the Terminals		
Terminal Name	Signal Type	Function
IN	N-bit input	data input port
OUT	N-bit output	positive logic data output port
SCANIN	input	scanpath serial input
SCANOUT	output	scanpath serial output
LOAD	input	high to load the register
LOADINV	input	low to load the register
SHIFT	input	test clock (master)
SHIFTINV	input	inverted test clock (master)
PHI1	input	slave clock
PHI1INV	input	inverted slave clock
Vdd	power	connects to power supply
GND	ground	connects to ground

Behavioral Description

```
/* Data is input through the datapath. */
   if ((LOAD == 1) && (LOADINV == 0)) {
      /*
       * Check to see if Scan path is active,
       * if not set REG to DATAIN.
       */
      if ((SHIFT == 1) && (SHIFTINV == 0))
         ERRMESS("Both Scanpath and Datapath are active.\n");
      else
         fcopy(REGm, !N!-1, 0, IN, !N!-1, 0);
      }
```


CNTLOUT1
CNTL1
CNTL2
CNTLOUT2
OUTINV[N-1]
IN[N-1]
N-1
N-1
OUTINV[0..N-2]
N-1
IN[0..N-2]
saturator

```
(end-sdl)
```

Performance Results

Leaf cell Delay Models				
Leaf cell	From	To	Best Fit Linear Model	Root Mean Square Error
saturator	IN, CNIL1	OUT INV	$t_{plh} = 1.2939 + 14.7864 * C_{load}$	0.0281
			$t_{phl} = 0.6433 + 6.5017 * C_{load}$	0.0095
			$t_{lh} = 2.2777 + 34.4291 * C_{load}$	0.0589
			$t_{hl} = 1.1236 + 13.3160 * C_{load}$	0.3682
	CNIL2	OUT INV	$t_{plh} = 1.0783 + 11.1602 * C_{load}$	0.0776
			$t_{phl} = 0.6152 + 4.1559 * C_{load}$	0.0236
			$t_{lh} = 1.8415 + 25.6985 * C_{load}$	0.2064
			$t_{hl} = 1.2355 + 7.8532 * C_{load}$	0.3482
saturator_nsb	IN	OUT INV	$t_{plh} = 2.7789 + 27.4036 * C_{load}$	0.7240
			$t_{phl} = 2.0265 + 28.4620 * C_{load}$	0.1539
			$t_{lh} = 2.1177 + 20.2924 * C_{load}$	0.7618
			$t_{hl} = 2.4120 + 16.0322 * C_{load}$	0.3167
		CNILOUT1	$t_{plh} = 0.7558 + 7.7772 * C_{load}$	0.0341
			$t_{phl} = 0.5032 + 4.1603 * C_{load}$	0.0216
			$t_{lh} = 1.1300 + 17.9220 * C_{load}$	0.0753
			$t_{hl} = 1.0254 + 7.8899 * C_{load}$	0.3529
		CNILOUT2	$t_{plh} = 1.6138 + 20.5987 * C_{load}$	0.0698
			$t_{phl} = 1.9882 + 16.0872 * C_{load}$	0.8889
			$t_{lh} = 2.4163 + 35.0500 * C_{load}$	0.2205
			$t_{hl} = 2.1046 + 16.6711 * C_{load}$	0.5075
	CNIL1	OUT INV	$t_{plh} = 1.3769 + 17.5250 * C_{load}$	0.0788
			$t_{phl} = 1.7010 + 22.6941 * C_{load}$	0.2154
			$t_{lh} = 1.6386 + 19.8599 * C_{load}$	0.1917
			$t_{hl} = 1.9422 + 16.4149 * C_{load}$	0.4032
		CNILOUT2	$t_{plh} = 1.4173 + 14.7807 * C_{load}$	0.0157
			$t_{phl} = 0.7033 + 6.5017 * C_{load}$	0.0095
			$t_{lh} = 2.8710 + 34.3698 * C_{load}$	0.3200
			$t_{hl} = 1.3720 + 13.2712 * C_{load}$	0.3258
	CNIL2	OUT INV	$t_{plh} = 1.2283 + 13.8822 * C_{load}$	0.2416
			$t_{phl} = 1.8225 + 22.7293 * C_{load}$	0.2070
			$t_{lh} = 0.9783 + 18.8748 * C_{load}$	0.3376
			$t_{hl} = 1.9595 + 16.3566 * C_{load}$	0.3830
		CNILOUT2	$t_{plh} = 1.5301 + 14.8097 * C_{load}$	0.0330
			$t_{phl} = 0.6473 + 4.1514 * C_{load}$	0.0257
			$t_{lh} = 2.8728 + 34.3656 * C_{load}$	0.3358
			$t_{hl} = 1.3826 + 7.8316 * C_{load}$	0.3289
Note: delays are in ns, C_{load} is in pF				

18 saturator

Synopsis

This block realizes the accumulator saturation feature in the DSPs designed using Lager. Basically, CNTL1 and CNTL2 indicate overflow or underflow conditions. More specifically, CNTL2 corresponds to an underflow and forces the N-1 least significant output bits to zero while generating a one in the most significant output bit. If CNTL1 and CNTL2 are both low then an overflow has occured and the most significant output bit is cleared while the remaining bits are set. In both cases, the msb behaves differently from the remaining bits since it acts as the sign bit. Finally, if CNTL1 is high and CNTL2 is low, the inverted input bits are passed to the output. Typically, this block sits in between the adder output and the accumulator input. It uses the leafcells *saturator* and *saturator_msb*.

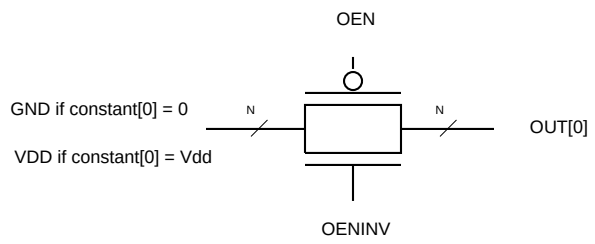
Description of the Terminals		
Terminal Name	Signal Type	Function
IN	N-bit input	data input
OUTINV	N-bit output	data output
CNTL1	input	control signal
CNTL2	input	control signal
CNTLOUT1	output	not(msb of IN)
CNTLOUT2	output	not(msb of OUTINV)
Vdd	power	connects to power supply
GND	ground	connects to ground

Behavioral Description

```
for (i=0;i<=N-2;i++) OUTINV[i]=nor(CNTL2,and(CNTL1,IN[i]));
CNTLOUT1=not(IN[N-1]);
CNTLOUT2=nor(CNTL2,and(CNTL1,CNTLOUT1));
OUTINV[N-1]=not(CNTLOUT2);
```

sdl File

```
(parent-cell saturator)
(parameters N)
(layout-generator TimLager)
(terminal CNTL1 (side BOT TOP ))
(terminal CNTL2 (side BOT TOP ))
(terminal CNTLOUT1 (side TOP))
(terminal CNTLOUT2 (side TOP))
(terminal GND (side BOT TOP ))
(terminal IN (side RIGHT LEFT ))
(terminal OUTINV (side RIGHT LEFT ))
(terminal Vdd (side BOT TOP ))
```



regconstant

Performance Results

Leafcell Delay Models				
Leafcell	From	To	Best Fit Linear Model	Root Mean Square Error
regone	CEN, CENINV	OUT	$t_{plh} = 0.3739 + 1.2474 * C_{load}$	0.0552
			$t_{phl} = 0.0000 + 0.0000 * C_{load}$	0.0000
			$t_{lh} = 1.0344 + 4.3811 * C_{load}$	0.1579
			$t_{hl} = 0.0000 + 0.0000 * C_{load}$	0.0000
regzero	CEN, CENINV	OUT	$t_{plh} = 0.0000 + 0.0000 * C_{load}$	0.0000
			$t_{phl} = 0.2488 + 1.0719 * C_{load}$	0.0799
			$t_{lh} = 0.0000 + 0.0000 * C_{load}$	0.0000
			$t_{hl} = 0.8245 + 2.4151 * C_{load}$	0.4072
Note: delays are in ns, C_{load} is in pF				

17 regconstant

Synopsis

This block implements a hardwired constant value which is enabled on the output, OUT, when OEN is true. It uses the leafcells *regone* and *regzero*.

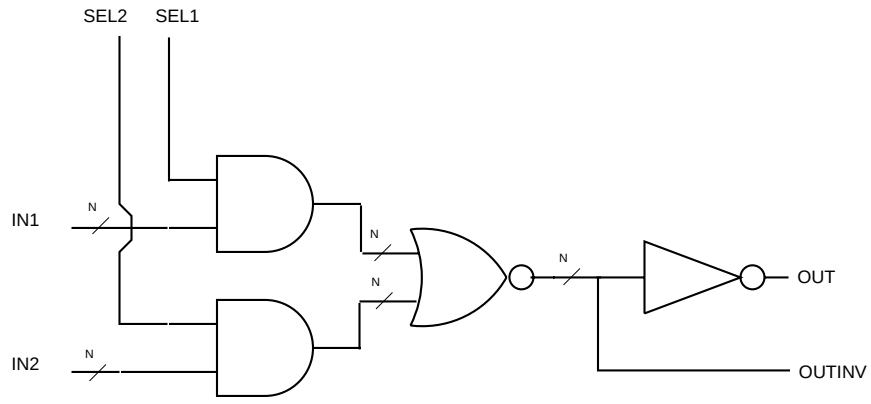
Description of the Terminals		
Terminal Name	Signal Type	Function
OUT	N-bit output	output data
OEN	input	high to read constant
Vdd	power	connects to power supply
GND	ground	connects to ground

Thor model

```
if(OEN==ONE) funpack(OUT,!N!-1,0,!constant!);
else fsetword(OUT,!N!-1,0,FLOAT);
if(!N! > sizeof(int)*8) fsetword(OUT,!N!-1,0,UNDEF);
if(OEN==OENINV) fsetword(OUT,!N!-1,0,UNDEF);
EXITMOD(0);
```

sdl File

```
(parent-cell regconstant)
(parameters N constant (MODULE_TYPE REGISTER))
(layout-generator TimLager)
(terminal OUT (side LEFT RIGHT)(TERMTYPE DATA_SIGNAL)(DIRECTION OUTPUT))
(terminal OEN    (side TOP BOT)(TERMTYPE CONTROL_SIGNAL)(DIRECTION INPUT))
(terminal OENINV (side TOP BOT)(TERMTYPE CONTROL_SIGNAL)(DIRECTION INPUT))
(terminal Vdd (side TOP BOT)(TERMTYPE SUPPLY))
(terminal GND (side TOP BOT)(TERMTYPE GROUND))
(end-sdl)
```



mux2to1

Performance Results

Leaf cell Delay Models				
Leaf cell	From	To	Best Fit Linear Model	Root Mean Square Error
mux2to1	IN1, SEL1	OUT	$t_{plh} = 1.3938 + 9.7597 * C_{load}$	0.1718
			$t_{phl} = 1.0619 + 8.3519 * C_{load}$	0.0708
			$t_{lh} = 1.6674 + 9.7074 * C_{load}$	0.8144
			$t_{hl} = 1.5358 + 6.8919 * C_{load}$	0.8513
		OUTINV	$t_{plh} = 0.8771 + 4.3394 * C_{load}$	0.0186
			$t_{phl} = 0.9480 + 4.2435 * C_{load}$	0.0568
			$t_{lh} = 1.6681 + 9.9311 * C_{load}$	0.1794
			$t_{hl} = 1.5040 + 8.7887 * C_{load}$	0.2844
	IN2, SEL2	OUT	$t_{plh} = 1.0755 + 9.7312 * C_{load}$	0.1298
			$t_{phl} = 0.8284 + 7.5552 * C_{load}$	0.0351
			$t_{lh} = 1.3407 + 9.8091 * C_{load}$	1.0677
			$t_{hl} = 1.1901 + 6.7023 * C_{load}$	0.9826
		OUTINV	$t_{plh} = 0.6674 + 3.6684 * C_{load}$	0.0632
			$t_{phl} = 0.6841 + 4.2351 * C_{load}$	0.0483
			$t_{lh} = 1.0804 + 8.6410 * C_{load}$	0.2791
			$t_{hl} = 1.2287 + 8.7305 * C_{load}$	0.2476
Note: delays are in ns, C_{load} is in pF				

16 mux2to1

Synopsis

This block implements a 2-to-1 multiplexer with both positive and negative logic outputs. Actually, it is slightly more general than a multiplexer. It has two control inputs, SEL1 and SEL2. When SEL1 and SEL2 are complementary, the block behaves like a multiplexer; otherwise, the block realizes an N-bit sum-of-products function in both positive and negative logic. Specifically, if neither SEL1 nor SEL2 is asserted, then OUT is 1; if only one of them is asserted, then OUT gets the value of the corresponding input (IN1 for SEL1 and IN2 for SEL2); finally, if both SEL1 and SEL2 are asserted, then OUT is the logical-OR of the two inputs. In all cases OUTINV is the inverse of OUT. This block uses the leafcell *mux2to1*.

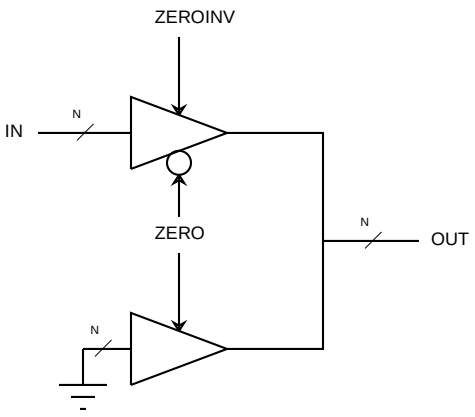
Description of the Terminals		
Terminal Name	Signal Type	Function
IN1	N-bit input	first input
IN2	N-bit input	second input
OUT	N-bit output	positive logic output port
OUTINV	N-bit output	negative logic output port
SEL1	input	high to select IN1
SEL2	input	high to select IN2
Vdd	power	connects to power supply
GND	ground	connects to ground

Behavioral Description

```
for (i=0;i<=N-1;i++) {
    OUTINV[i]=nor(and(SEL1,IN1[i]),and(SEL2,IN2[i]));
    OUT[i]=not(OUTINV[i]);
    }
```

sdl File

```
(parent-cell mux2to1)
(parameters N)
(layout-generator TimLager)
(terminal SEL1 (side TOP BOT ))
(terminal SEL2 (side TOP BOT ))
(terminal GND (side TOP BOT ))
(terminal IN1 (side RIGHT LEFT ))
(terminal IN2 (side RIGHT LEFT ))
(terminal OUT (side RIGHT ))
(terminal OUTINV (side LEFT ))
(terminal Vdd (side TOP BOT ))
(end-sdl)
```



isozero

Performance Results

Leafcell Delay Models				
Leafcell	From	To	Best Fit Linear Model	Root Mean Square Error
isozero	ZERO	OUT	$t_{plh} = 0.4747 + 3.8588 * C_{load}$	0.0711
			$t_{phl} = 0.3860 + 4.1356 * C_{load}$	0.0168
			$t_{lh} = 0.6270 + 14.3737 * C_{load}$	0.2252
			$t_{hl} = 0.4093 + 7.9129 * C_{load}$	0.1993
Note: delays are in ns, C_{load} is in pF				

15 isozero

Synopsis

This block effectively acts as a 2-to-1 multiplexer whose one input is zero. Basically, if ZERO and ZEROINV are asserted 1 and 0 respectively, the output is forced to be all zeros. otherwise, output is the same as the input. It uses the leafcell *isozero*.

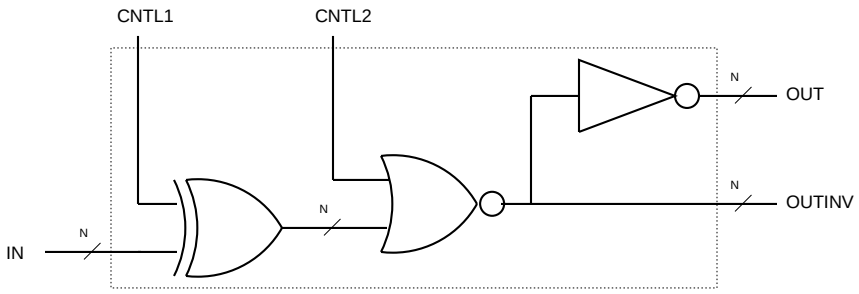
Description of the Terminals		
Terminal Name	Signal Type	Function
IN	N-bit input	input data port
OUT	N-bit output	output data port
ZERO	input	high for output to be all zeros
Vdd	power	connects to power supply
GND	ground	connects to ground

Behavioral Description

```
if (ZERO==ZEROINV) ERROR();
for (i=0;i<=N-1;i++) {
    if (ZERO==1) OUT[i]=0;
    else OUT[i]=IN[i];
    }
```

sdl File

```
(parent-cell isozero)
(parameters N)
(layout-generator TimLager)
(terminal GND (side TOP BOT ))
(terminal IN (side RIGHT LEFT ))
(terminal OUT (side RIGHT LEFT ))
(terminal Vdd (side TOP BOT ))
(terminal ZERO (side TOP BOT ))
(terminal ZEROINV (side TOP BOT ))
(end-sdl)
```



OUT[i]=not(outinv[i])

OUTINV[i]=not (CNTL2 or (CNTL1 xor IN[i]))

isoinvzero

Performance Results

Leafcell Delay Models				
Leafcell	From	To	Best Fit Linear Model	Root Mean Square Error
isoinvzero	IN	OUT	$t_{plh} = 1.9860 + 4.6209 * C_{load}$	0.1343
			$t_{phl} = 1.4627 + 9.1266 * C_{load}$	0.0077
			$t_{lh} = 0.3438 + 4.7744 * C_{load}$	0.0625
			$t_{hl} = 0.4048 + 6.9368 * C_{load}$	0.0373
isoinvzero	IN	OUTINV	$t_{plh} = 1.2302 + 5.0193 * C_{load}$	0.0127
			$t_{phl} = 1.8202 + 2.1193 * C_{load}$	0.1377
			$t_{lh} = 1.1718 + 11.8422 * C_{load}$	0.0492
			$t_{hl} = 0.8831 + 3.7970 * C_{load}$	0.0676
isoinvzero	CNIL2	OUT	$t_{plh} = 0.6292 + 4.5423 * C_{load}$	0.0192
			$t_{phl} = 0.9394 + 9.1006 * C_{load}$	0.0316
			$t_{lh} = 0.2873 + 4.7977 * C_{load}$	0.0573
			$t_{hl} = 0.4048 + 6.9368 * C_{load}$	0.0373
isoinvzero	CNIL2	OUTINV	$t_{plh} = 0.7040 + 4.9961 * C_{load}$	0.0234
			$t_{phl} = 0.4782 + 2.0335 * C_{load}$	0.0191
			$t_{lh} = 1.1898 + 11.8343 * C_{load}$	0.0959
			$t_{hl} = 0.6938 + 3.8207 * C_{load}$	0.1486
Note: delays are in ns, C_{load} is in pF				

14 isoinvzero

Synopsis

This block implements the logic which, depending on the value of the control signals CNTL1 and CNTL2, either just passes the input IN to the output OUTINV, or passes the logical inverse of the input or makes the output all zeros. The output OUT is a logical inverse of OUTINV. The leafcell used is *isoinvzero*.

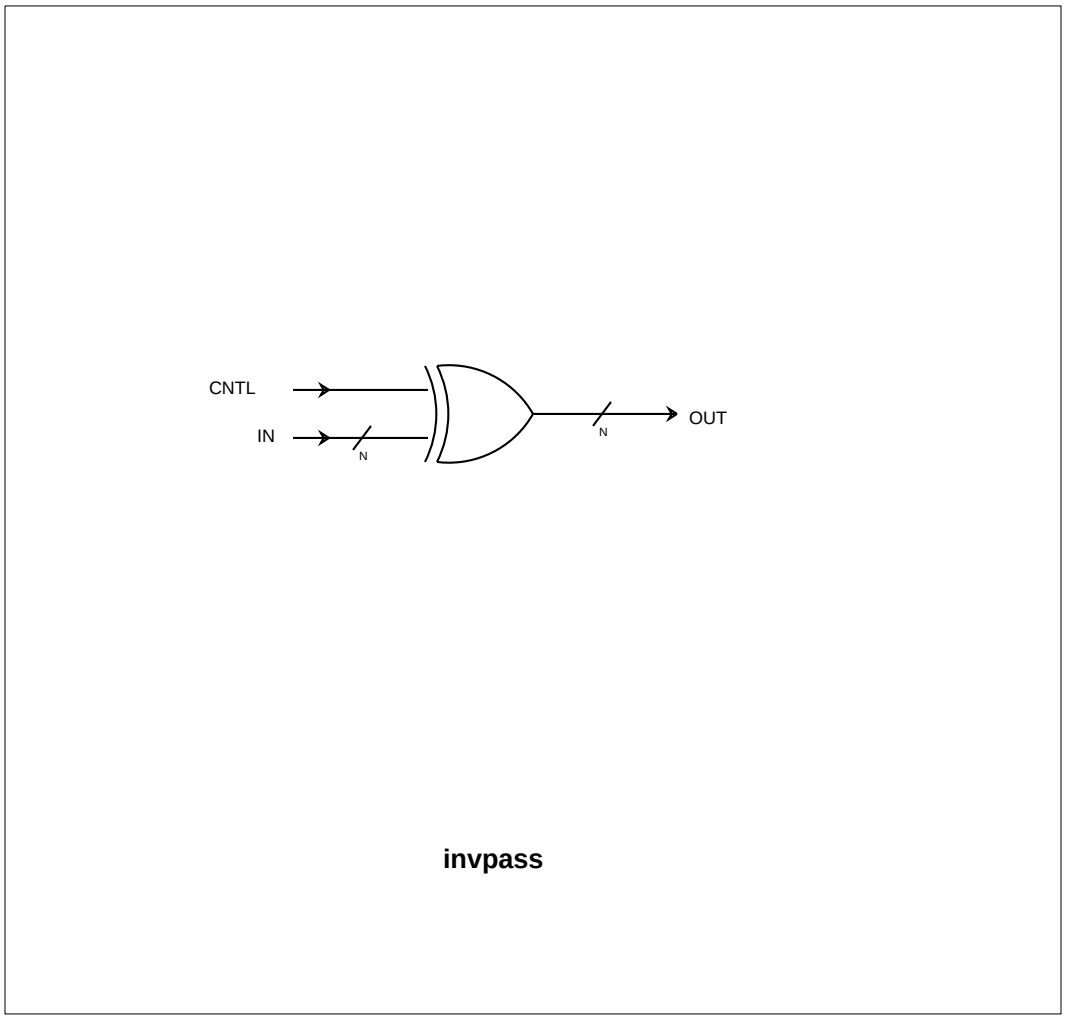
Description of the Terminals		
Terminal Name	Signal Type	Function
IN	N-bit input	data input port
OUTINV	N-bit output	data output port
OUT	N-bit output	data output port
CNTL1	input	control signal to complement
CNTL2	input	control signal to zero the output
Vdd	power	connects to power supply
GND	ground	connects to ground

Behavioral Description

```
if (CNTL2==1)
    for (i=0;i<=N-1;i++) OUTINV[i]=0;
else if (CNTL1==0)
    for (i=0;i<=N-1;i++) OUTINV[i]=not(IN[i]);
else    /*(CNTL1==1) && (CNTL2==0)*/
    for (i=0;i<=N-1;i++) OUTINV[i]=IN[i];
for (i=0;i<=N-1;i++) OUT[i]=not(OUTINV[i]);
```

sdl File

```
(parent-cell isoinvzero)
(parameters N)
(layout-generator TimLager)
(terminal CNTL1 (side BOT TOP ))
(terminal CNTL2 (side TOP BOT ))
(terminal GND (side TOP BOT ))
(terminal IN (side LEFT))
(terminal OUTINV (side RIGHT))
(terminal OUT (side RIGHT))
(terminal Vdd (side BOT TOP ))
(end-sdl)
```


CNTL
IN
N
N
OUT
invpass

Performance Results

Leafcell Delay Models				
Leafcell	From	To	Best Fit Linear Model	Root Mean Square Error
invpass	IN	OUT	$t_{plh} = 0.8182 + 5.4725 * C_{load}$	0.0299
			$t_{phl} = 0.4382 + 2.3188 * C_{load}$	0.0174
			$t_{lh} = 1.6205 + 12.8434 * C_{load}$	0.1337
			$t_{hl} = 1.0290 + 4.8059 * C_{load}$	0.1562
	CNTL	OUT	$t_{plh} = 0.8221 + 5.4809 * C_{load}$	0.0216
			$t_{phl} = 0.4775 + 2.3244 * C_{load}$	0.0301
			$t_{lh} = 1.3456 + 12.8262 * C_{load}$	0.1241
			$t_{hl} = 0.9835 + 4.7990 * C_{load}$	0.0726
Note: delays are in ns, C_{load} is in pF				

13 invpass

Synopsis

This block passes either the input or the complement of the input to the output. In other words, every output bit is the XOR of the corresponding input bit with the control signal. It uses the leafcell *invpass* in all the bit positions.

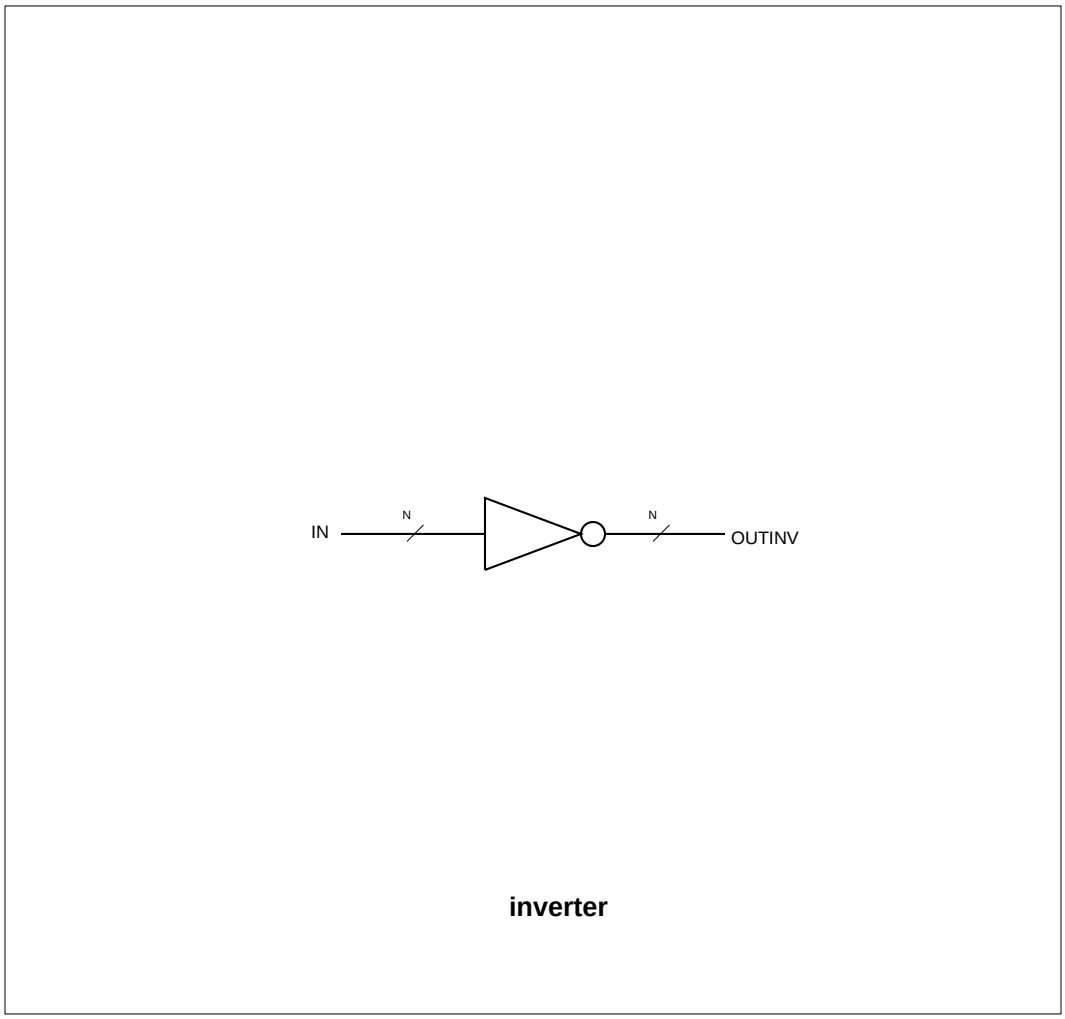
Description of the Terminals		
Terminal Name	Signal Type	Function
IN	N-bit input	input
OUT	N-bit output	output
CNTL	input	control input
Vdd	power	connects to power supply
GND	ground	connects to ground

Behavioral Description

```
for (i=0;i<=N-1;i++) {
    if (CNTL==0) {
        OUT[i] = IN[i]
    } else {
        OUT[i] = !(IN[i])
    }
}
```

sdl File

```
(parent-cell invpass)
(parameters
    N
    (FEEDTHRUS 0)
    (BITHEIGHT 0)
    )
(layout-generator TimLager)
(structure-processor dpp)
(terminal GND (side TOP BOT))
(terminal Vdd (side TOP BOT))
(terminal OUT (side LEFT RIGHT))
(terminal IN  (side LEFT))
(terminal CNTL (side TOP BOT))
(end-sdl)
```


IN
N
N
OUTINV
inverter

12 inverter

Synopsis

This blocks implements a N-bit inverter. It uses the leafcell *inverter*.

Description of the Terminals		
Terminal Name	Signal Type	Function
IN	N-bit input	input data port
OUTINV	N-bit output	output data port
Vdd	power	connects to power supply
GND	ground	connects to ground

Behavioral Description

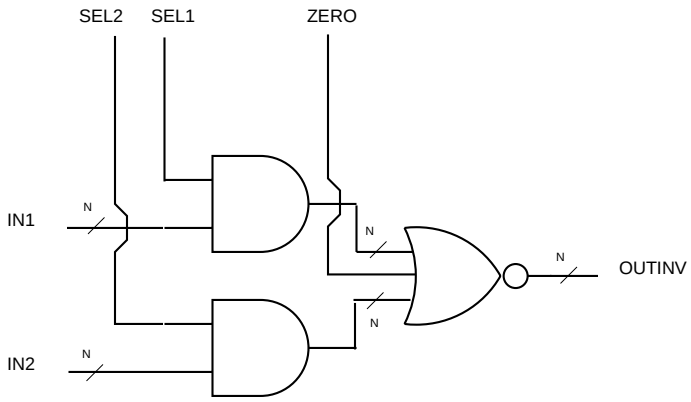
```
for (i=0;i<=N-1;i++) OUTINV[i]=not(IN[i]);
```

sdl File

```
(parent-cell inverter)
(parameters N)
(layout-generator TimLager)
(terminal GND (side TOP BOT ))
(terminal IN (side LEFT RIGHT ))
(terminal OUTINV (side RIGHT LEFT ))
(terminal Vdd (side TOP BOT ))
(end-sdl)
```

Performance Results

Leafcell Delay Models				
Leafcell	From	To	Best Fit Linear Model	Root Mean Square Error
inverter	IN	OUTINV	$t_{plh} = 0.4168 + 3.9324 * C_{load}$	0.0392
			$t_{phl} = 0.3617 + 4.1568 * C_{load}$	0.1375
			$t_{lh} = 0.3523 + 9.1246 * C_{load}$	0.6136
			$t_{hl} = 0.5397 + 7.9930 * C_{load}$	0.3051
Note: delays are in ns, C_{load} is in pF				



inv2to1muxzero

Performance Results

Leafcell Delay Models				
Leafcell	From	To	Best Fit Linear Model	Root Mean Square Error
inv2to1muxzero	IN1, SEL1	OUTINV	$t_{plh} = 1.5784 + 18.1308 * C_{load}$	0.0739
			$t_{phl} = 0.7636 + 6.5234 * C_{load}$	0.0084
			$t_{lh} = 2.7040 + 42.4791 * C_{load}$	0.4114
			$t_{hl} = 1.3275 + 13.3244 * C_{load}$	0.1171
	IN2, SEL2	OUTINV	$t_{plh} = 1.1871 + 18.2004 * C_{load}$	0.0218
			$t_{phl} = 0.5407 + 6.4944 * C_{load}$	0.0140
			$t_{lh} = 2.3716 + 42.6155 * C_{load}$	0.3951
			$t_{hl} = 0.9365 + 13.3083 * C_{load}$	0.3451
	ZERO	OUTINV	$t_{plh} = 1.1380 + 14.5458 * C_{load}$	0.0613
			$t_{phl} = 0.5499 + 4.1587 * C_{load}$	0.0249
			$t_{lh} = 1.7552 + 33.9998 * C_{load}$	0.1500
			$t_{hl} = 0.8471 + 7.9857 * C_{load}$	0.3082
Note: delays are in ns, C_{load} is in pF				

11 inv2to1muxzero

Synopsis

This block implements a 2-to-1 multiplexer with a negative logic output and a signal to force the output low. If ZERO is high, then the outputs are forced low; however, if ZERO is low, then the block acts as an enhanced multiplexer. Specifically, if the two control inputs, SEL1 and SEL2, are complementary, then the block behaves like a multiplexer; otherwise, it implements an N-bit sum-of-products function with a negative logic output. For example, if neither SEL1 nor SEL2 is asserted, the output is 1; if one of them is asserted, the output is the logical inverse of the corresponding input; and if both of them are asserted, the output is the logical-NOR of the two inputs. It uses the leafcell *inv2to1muxzero*.

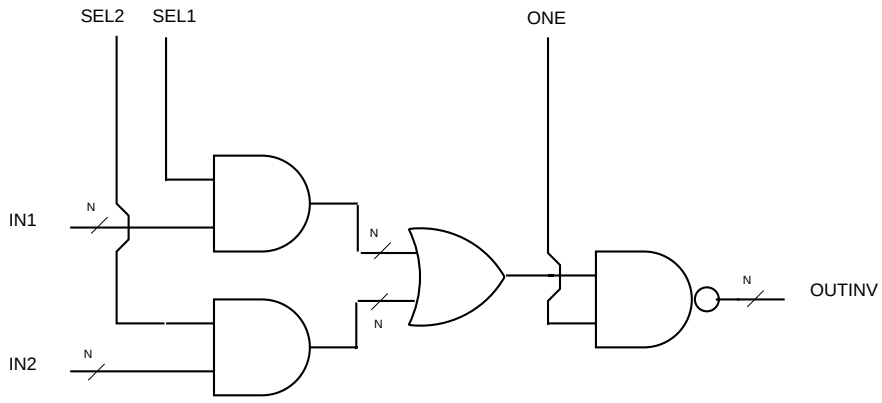
Description of the Terminals		
Terminal Name	Signal Type	Function
IN1	N-bit input	first input
IN2	N-bit input	second input
OUTINV	N-bit output	output port
SEL1	input	high to select IN1
SEL2	input	high to select IN2
ZERO	input	high for output to be all zeros
Vdd	power	connects to power supply
GND	ground	connects to ground

Behavioral Description

```
for (i=0;i<=N-1;i++) {
    if (ZERO==1) OUTINV[i]=0;
    else OUTINV[i]=nor(and(SEL1,IN1[i]),and(SEL2,IN2[i]));
    }
```

sdl File

```
(parent-cell inv2to1muxzero)
(parameters N)
(layout-generator TimLager)
(terminal GND (side TOP BOT ))
(terminal IN1 (side LEFT RIGHT ))
(terminal IN2 (side LEFT RIGHT ))
(terminal OUTINV (side LEFT RIGHT ))
(terminal SEL1 (side TOP BOT ))
(terminal SEL2 (side TOP BOT ))
(terminal Vdd (side TOP BOT ))
(terminal ZERO (side TOP BOT ))
(end-sdl)
```



inv2to1muxone

Performance Results

Leafcell Delay Models				
Leafcell	From	To	Best Fit Linear Model	Root Mean Square Error
inv2to1muxone	IN1, SEL1	OUTINV	$t_{plh} = 1.1380 + 9.2288 * C_{load}$	0.0336
			$t_{phl} = 0.8872 + 7.0637 * C_{load}$	0.0552
			$t_{lh} = 1.8313 + 21.3915 * C_{load}$	0.3715
			$t_{hl} = 1.5800 + 15.0610 * C_{load}$	0.0904
	IN2, SEL2	OUTINV	$t_{plh} = 1.0044 + 10.5348 * C_{load}$	0.0374
			$t_{phl} = 0.6296 + 7.0663 * C_{load}$	0.0338
			$t_{lh} = 1.7846 + 24.3101 * C_{load}$	0.2153
			$t_{hl} = 1.1773 + 15.0977 * C_{load}$	0.1307
	ONE	OUTINV	$t_{plh} = 0.7967 + 7.7667 * C_{load}$	0.0490
			$t_{phl} = 0.7786 + 7.0599 * C_{load}$	0.0273
			$t_{lh} = 1.5315 + 17.8741 * C_{load}$	0.1228
			$t_{hl} = 1.5715 + 15.0595 * C_{load}$	0.1038
Note: delays are in ns, C_{load} is in pF				

10 inv2to1muxone

Synopsis

This block implements a 2-to-1 multiplexer with a negative logic output and a signal to force the output high. If ONE is low, then the outputs are forced high; however, if ONE is high, then the block acts as an enhanced multiplexer. Specifically, if the two control inputs, SEL1 and SEL2, are complementary, then the block behaves like a multiplexer; otherwise, it implements an N-bit sum-of-products function with a negative logic output. For example, if neither SEL1 nor SEL2 is asserted, the output is 1; if one of them is asserted, the output is the logical inverse of the corresponding input; and if both of them are asserted, the output is the logical-NOR of the two inputs. It uses the leafcell *inv2to1muxone*.

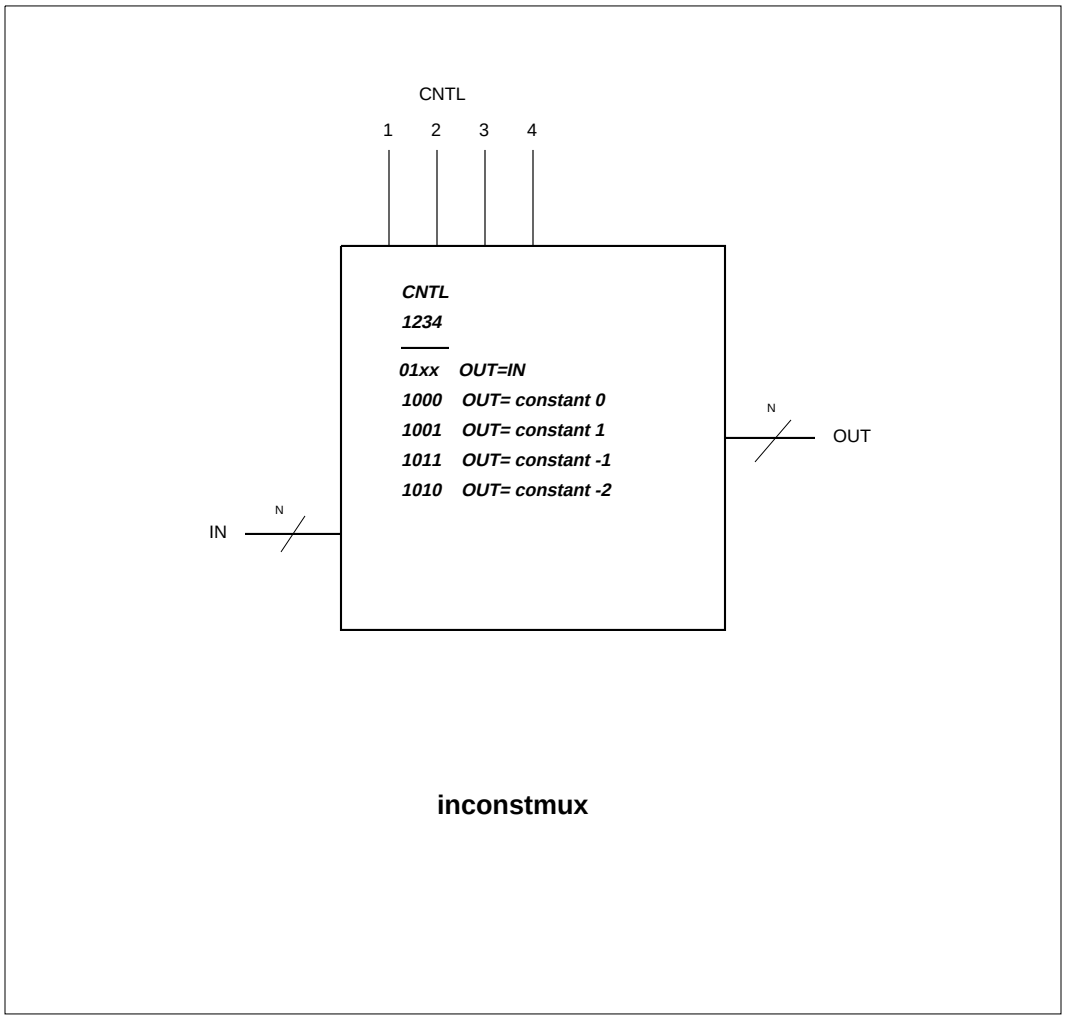
Description of the Terminals		
Terminal Name	Signal Type	Function
IN1	N-bit input	first input
IN2	N-bit input	second input
OUTINV	N-bit output	output port
SEL1	input	high to select IN1
SEL2	input	high to select IN2
ONE	input	low for output to be all ones
Vdd	power	connects to power supply
GND	ground	connects to ground

Behavioral Description

```
for (i=0;i<=N-1;i++) {
    if (ONE==0) OUTINV[i]=1;
    else OUTINV[i]=nor(and(SEL1,IN1[i]),and(SEL2,IN2[i]));
    }
```

sdl File

```
(parent-cell inv2to1muxone)
(parameters N)
(layout-generator TimLager)
(terminal GND (side TOP BOT ))
(terminal IN1 (side LEFT RIGHT ))
(terminal IN2 (side LEFT RIGHT ))
(terminal ONE (side BOT TOP ))
(terminal OUTINV (side LEFT RIGHT ))
(terminal SEL1 (side TOP BOT ))
(terminal SEL2 (side TOP BOT ))
(terminal Vdd (side TOP BOT ))
(end-sdl)
```


CNTL
1 2 3 4
CNTL
1234
01xx OUT=IN
1000 OUT= constant 0
1001 OUT= constant 1
1011 OUT= constant -1
1010 OUT= constant -2
N
OUT
N
IN
inconstmux

Performance Results

Leafcell Delay Models				
Leafcell	From	To	Best Fit Linear Model	Root Mean Square Error
inconmx_lsb	CNIL1	OUT	$t_{plh} = 1.1762 + 3.4866 * C_{load}$	0.0091
			$t_{phl} = 1.1306 + 2.3604 * C_{load}$	0.0694
			$t_{lh} = 0.3725 + 8.0683 * C_{load}$	0.1023
			$t_{hl} = 0.3834 + 4.4357 * C_{load}$	0.0959
inconmx_lsb	CNIL4	OUT	$t_{plh} = 1.2057 + 3.4822 * C_{load}$	0.0056
			$t_{phl} = 1.1560 + 2.3576 * C_{load}$	0.0627
			$t_{lh} = 0.3787 + 8.0622 * C_{load}$	0.0953
			$t_{hl} = 0.3788 + 4.4329 * C_{load}$	0.1025
inconmx_lsb	IN	OUT	$t_{plh} = 0.8836 + 3.4867 * C_{load}$	0.0038
			$t_{phl} = 0.8644 + 2.3348 * C_{load}$	0.0347
			$t_{lh} = 0.3121 + 8.0882 * C_{load}$	0.0703
			$t_{hl} = 0.2920 + 4.4639 * C_{load}$	0.1048
inconmx_other	CNIL1	OUT	$t_{plh} = 1.1762 + 3.4866 * C_{load}$	0.0091
			$t_{phl} = 1.1306 + 2.3604 * C_{load}$	0.0694
			$t_{lh} = 0.3725 + 8.0683 * C_{load}$	0.1023
			$t_{hl} = 0.3834 + 4.4357 * C_{load}$	0.0959
inconmx_other	CNIL3	OUT	$t_{plh} = 1.2057 + 3.4822 * C_{load}$	0.0056
			$t_{phl} = 1.1560 + 2.3576 * C_{load}$	0.0627
			$t_{lh} = 0.3787 + 8.0622 * C_{load}$	0.0953
			$t_{hl} = 0.3788 + 4.4329 * C_{load}$	0.1025
inconmx_other	IN	OUT	$t_{plh} = 0.8836 + 3.4867 * C_{load}$	0.0038
			$t_{phl} = 0.8644 + 2.3348 * C_{load}$	0.0347
			$t_{lh} = 0.3121 + 8.0882 * C_{load}$	0.0703
			$t_{hl} = 0.2920 + 4.4639 * C_{load}$	0.1048
Note: delays are in ns, C_{load} is in pF				

sdl File

```
(parent-cell inconstmux)
(parameters N (FEEDTHRUS 0) (BITHEIGHT 0) (MODULE_TYPE "MUX"))
(layout-generator TimLager)
(structure-processor dpp)
(terminal IN (TERM_EDGE LEFT) (TERM_EDGE RIGHT)(TERMTYPE DATA_SIGNAL)(DIRECTION
INPUT))
(terminal OUT (TERM_EDGE RIGHT)(TERMTYPE DATA_SIGNAL)(DIRECTION OUTPUT))
(terminal CNTL1 (TERM_EDGE TOP) (TERM_EDGE BOTTOM)(TERMTYPE CONTROL_SIGNAL)(DIRECTION
INPUT))
(terminal CNTL2 (TERM_EDGE TOP) (TERM_EDGE BOTTOM)(TERMTYPE CONTROL_SIGNAL)(DIRECTION
INPUT))
(terminal CNTL3 (TERM_EDGE TOP) (TERM_EDGE BOTTOM)(TERMTYPE CONTROL_SIGNAL)(DIRECTION
INPUT))
(terminal CNTL4 (TERM_EDGE TOP) (TERM_EDGE BOTTOM)(TERMTYPE CONTROL_SIGNAL)(DIRECTION
INPUT))
(terminal GND (TERM_EDGE TOP) (TERM_EDGE BOTTOM)(TERMTYPE GROUND))
(terminal Vdd (TERM_EDGE TOP) (TERM_EDGE BOTTOM)(TERMTYPE SUPPLY))
(end-sdl)
```

9 inconstmux

Synopsis

This block can be used to pass the input data, IN, to the output, OUT, or select one of four constants (0, 1, -1, -2) to put on the output. This function is very useful in address calculation units. The appropriate values of the control signals are as follows:

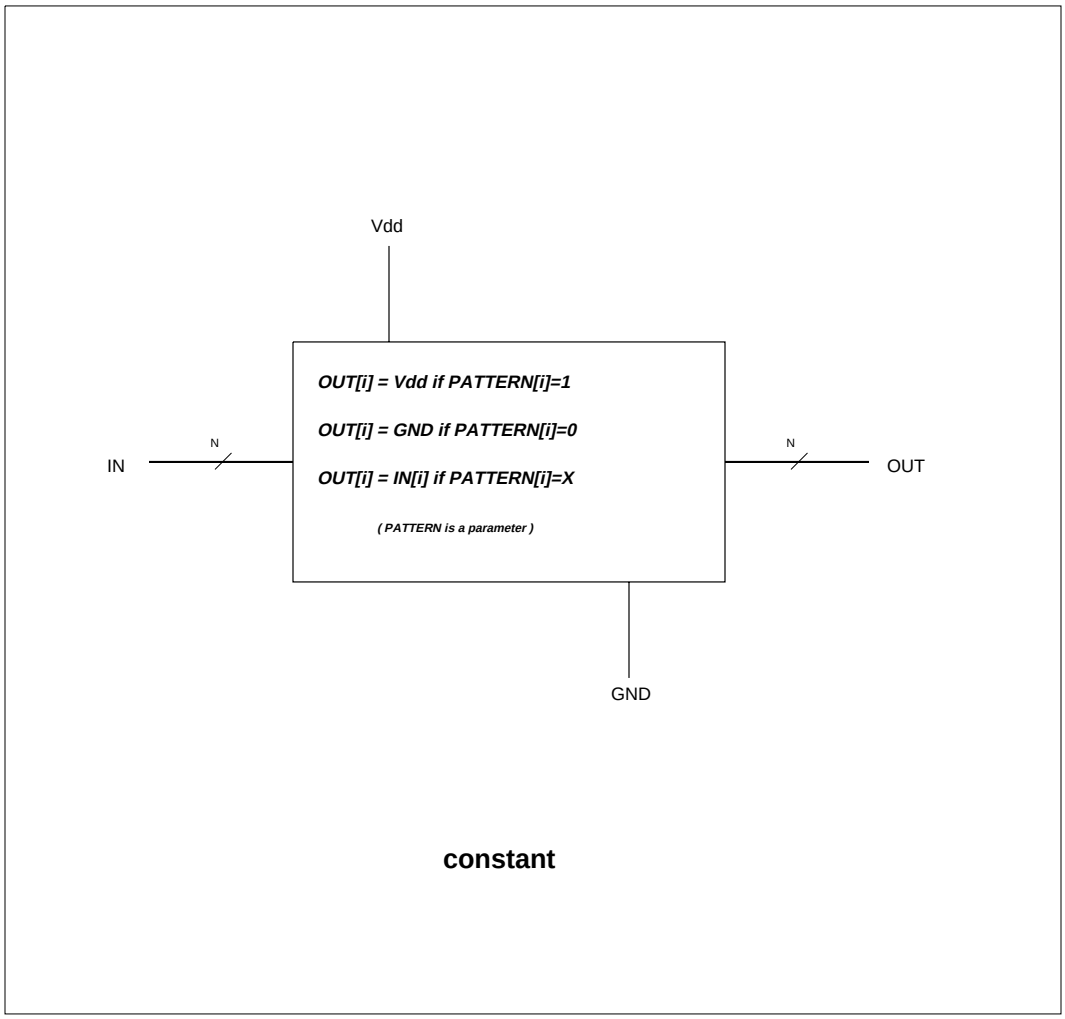
CNTL 1234	Result
01xx	pass input data
1000	constant 0
1001	constant 1
1011	constant -1
1010	constant -2

Other combinations of control signals are invalid. It uses the leafcells *inconmx_lsb* and *inconmx_other*.

Description of the Terminals		
Terminal Name	Signal Type	Function
IN	N-bit input	input data
OUT	N-bit output	output data
CNTL1	input	control input
CNTL2	input	control input
CNTL3	input	control input
CNTL4	input	control input
Vdd	power	connects to power supply
GND	ground	connects to ground

Thor model

```
if(CNTL1==ONE){
  fsetword(OUT,!N!-1,0,ZERO);
  if(CNTL3==ONE) fsetword(OUT,!N!-1,1,ONE);
  if(CNTL4==ONE) OUT[0]=ONE;
}
else{
     fcopy(OUT,!N!-1,0,IN,!N!-1,0);
}
if(fsckbin(CNTL1)!=PASSED) fsetword(OUT,!N!-1,0,UNDEF);
if(fsckbin(CNTL2)!=PASSED) fsetword(OUT,!N!-1,0,UNDEF);
if(fsckbin(CNTL3)!=PASSED) fsetword(OUT,!N!-1,0,UNDEF);
if(fsckbin(CNTL4)!=PASSED) fsetword(OUT,!N!-1,0,UNDEF);
if(CNTL1==CNTL2) fsetword(OUT,!N!-1,0,UNDEF);
EXITMOD(0);
```


Vdd
OUT[i] = Vdd if PATTERN[i]=1
OUT[i] = GND if PATTERN[i]=0
OUT[i] = IN[i] if PATTERN[i]=X
(PATTERN is a parameter)
N
IN
N
OUT
GND

constant

```
(terminal OUT (side RIGHT))
(terminal GND (side TOP BOT ))
(terminal Vdd (side TOP BOT ))
(end-sdl)
```

8 constant

Synopsis

This block implements a constant value in a datapath. It ctually does a little bit more. The output bits of the block are either 0 (tied to GND), 1 (tied to Vdd) or equal to the corresponding input bit. A parameter called PATTERN is used to specify what each output bit is connected to. The value of PATTERN is a string of 0 or 1 or X which stand for connect to GND, connect to Vdd and connect to the input bit respectively. The first (leftmost) character of the string stands for the bit 0.

If PATETRN consists of only 0's and 1's then the block implements a constant. If X's are used too then it acts as if some of the bist are constant.

The block uses the leafcells *constIn*, *constOne* and *constZero*.

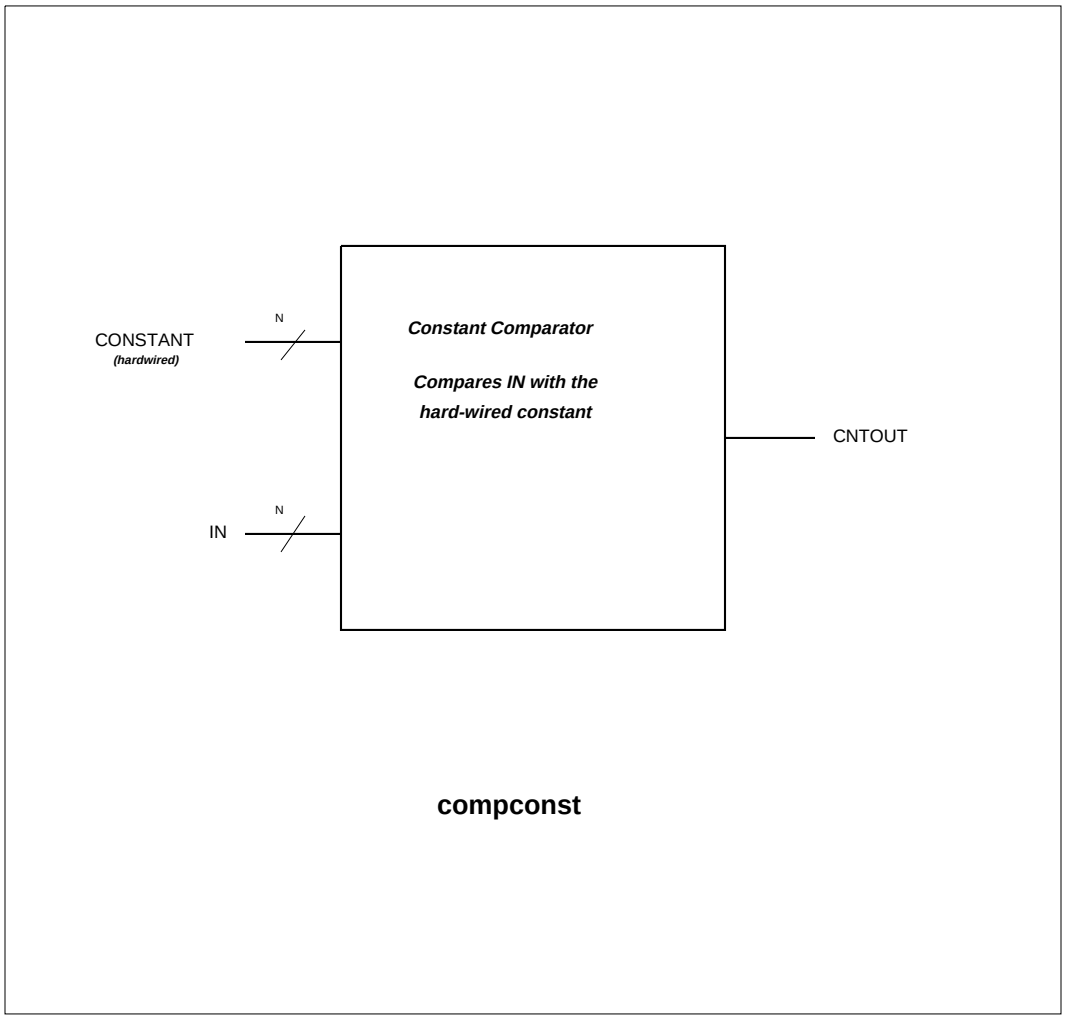
Description of the Terminals		
Terminal Name	Signal Type	Function
IN	N-bit input	input (may be unconnected)
OUT	N-bit output	output
Vdd	power	connects to power supply
GND	ground	connects to ground

Behavioral Description

```
char *p = PATTERN;
for (i=0;i<N;i++) {
    if (p[i]='1')
        OUT[i]=Vdd;
    else if (p[i]='o')
        OUT[i]=GND;
    else
        OUT[i]=IN[i];
}
```

sdl File

```
(parent-cell constant)
(parameters
    N
    PATTERN
    (FEEDTHRUS 0)
    (BITHEIGHT 0)
    )
(layout-generator TimLager)
(structure-processor dpp)
(terminal IN (side LEFT))
```


CONSTANT
(hardwired)
N
IN
N
Constant Comparator
Compares IN with the
hard-wired constant
CNTOUT
compconst

Leafcell Delay Models				
Leafcell	From	To	Best Fit Linear Model	Root Mean Square Error
comconQ_Qnsb	IN	CNTOUT	$t_{plh} = 0.4350 + 6.1732 * C_{load}$	0.0263
			$t_{phl} = 0.8441 + 3.2911 * C_{load}$	0.0061
			$t_{lh} = 0.2683 + 14.4138 * C_{load}$	0.0290
			$t_{hl} = 0.1891 + 6.3399 * C_{load}$	0.0820
comconQ_Qnsb	CNTIN	CNTOUT	$t_{plh} = 0.4729 + 6.1776 * C_{load}$	0.0233
			$t_{phl} = 0.7516 + 3.2912 * C_{load}$	0.0040
			$t_{lh} = 0.2683 + 14.4138 * C_{load}$	0.0290
			$t_{hl} = 0.1784 + 6.3455 * C_{load}$	0.0682
comconQ_Z	IN	CNTOUT	$t_{plh} = 0.7120 + 4.4249 * C_{load}$	0.0152
			$t_{phl} = 0.7967 + 2.7983 * C_{load}$	0.0095
			$t_{lh} = 0.4200 + 10.3610 * C_{load}$	0.0342
			$t_{hl} = 0.2442 + 5.8471 * C_{load}$	0.0731
comconQ_Z	CNTIN	CNTOUT	$t_{plh} = 0.5026 + 4.4389 * C_{load}$	0.0355
			$t_{phl} = 0.5139 + 2.3937 * C_{load}$	0.2911
			$t_{lh} = 0.4371 + 10.2930 * C_{load}$	0.1626
			$t_{hl} = 0.4359 + 5.7723 * C_{load}$	0.2314
comconQ_Znsb	IN	CNTOUT	$t_{plh} = 0.4473 + 3.7977 * C_{load}$	0.0329
			$t_{phl} = 0.4927 + 2.1266 * C_{load}$	0.1633
			$t_{lh} = 0.5348 + 8.9905 * C_{load}$	0.1838
			$t_{hl} = 0.4113 + 4.4011 * C_{load}$	0.2041
comconQ_Znsb	CNTIN	CNTOUT	$t_{plh} = 0.7122 + 3.8222 * C_{load}$	0.0082
			$t_{phl} = 1.0487 + 2.3379 * C_{load}$	0.0182
			$t_{lh} = 0.3874 + 9.0611 * C_{load}$	0.0383
			$t_{hl} = 0.3967 + 4.4667 * C_{load}$	0.0980
Note: delays are in ns, C_{load} is in pF				

Performance Results

Leaf cell Delay Models				
Leaf cell	From	To	Best Fit Linear Model	Root Mean Square Error
comconE_O	IN	CNTOUT	$t_{plh} = 0.8584 + 3.8088 * C_{load}$	0.0040
			$t_{phl} = 0.9866 + 2.3350 * C_{load}$	0.0150
			$t_{lh} = 0.4128 + 9.0583 * C_{load}$	0.0452
			$t_{hl} = 0.4005 + 4.4727 * C_{load}$	0.1049
comconE_O	CNTIN	CNTOUT	$t_{plh} = 0.4527 + 3.7949 * C_{load}$	0.0399
			$t_{phl} = 0.4008 + 2.3334 * C_{load}$	0.0202
			$t_{lh} = 0.5548 + 8.9905 * C_{load}$	0.1838
			$t_{hl} = 0.4193 + 4.4056 * C_{load}$	0.2069
comconE_Z	IN	CNTOUT	$t_{plh} = 0.4422 + 3.7906 * C_{load}$	0.0422
			$t_{phl} = 0.4291 + 2.2179 * C_{load}$	0.0584
			$t_{lh} = 0.5268 + 8.9860 * C_{load}$	0.1812
			$t_{hl} = 0.4139 + 4.4011 * C_{load}$	0.1987
comconE_Z	CNTIN	CNTOUT	$t_{plh} = 0.3720 + 3.8005 * C_{load}$	0.0260
			$t_{phl} = 0.4840 + 2.2107 * C_{load}$	0.0502
			$t_{lh} = 0.4917 + 9.0032 * C_{load}$	0.1558
			$t_{hl} = 0.5234 + 4.3967 * C_{load}$	0.2014
comconE_Zlsb	IN	CNTOUT	$t_{plh} = 0.8705 + 5.2507 * C_{load}$	0.6169
			$t_{phl} = 0.4261 + 5.5989 * C_{load}$	0.0223
			$t_{lh} = 0.3684 + 14.3698 * C_{load}$	0.1802
			$t_{hl} = 0.3298 + 10.7490 * C_{load}$	0.1645
comconO_O	IN	CNTOUT	$t_{plh} = 0.5601 + 4.4243 * C_{load}$	0.0393
			$t_{phl} = 0.4166 + 2.3887 * C_{load}$	0.2051
			$t_{lh} = 0.5404 + 10.2947 * C_{load}$	0.1721
			$t_{hl} = 0.3816 + 5.8082 * C_{load}$	0.1843
comconO_O	CNTIN	CNTOUT	$t_{plh} = 0.5047 + 4.4345 * C_{load}$	0.0327
			$t_{phl} = 0.4825 + 2.3902 * C_{load}$	0.2019
			$t_{lh} = 0.4371 + 10.2930 * C_{load}$	0.1626
			$t_{hl} = 0.4321 + 5.7882 * C_{load}$	0.2167
Note: delays are in ns, C_{load} is in pF				

7 compconst

Synopsis

The status output signal, CNTOUT, is high whenever, the input, IN, matches the hardwired constant. Currently a datapath which uses this block cannot be wider than the c integer width of the machine on which it is compiled. It uses the leafcells *comconE_O comconE_Olsb comconE_Z comconE_Zlsb comconQ_O comconQ_Omsb comconQ_Z* and *comconQ_Zmsb*.

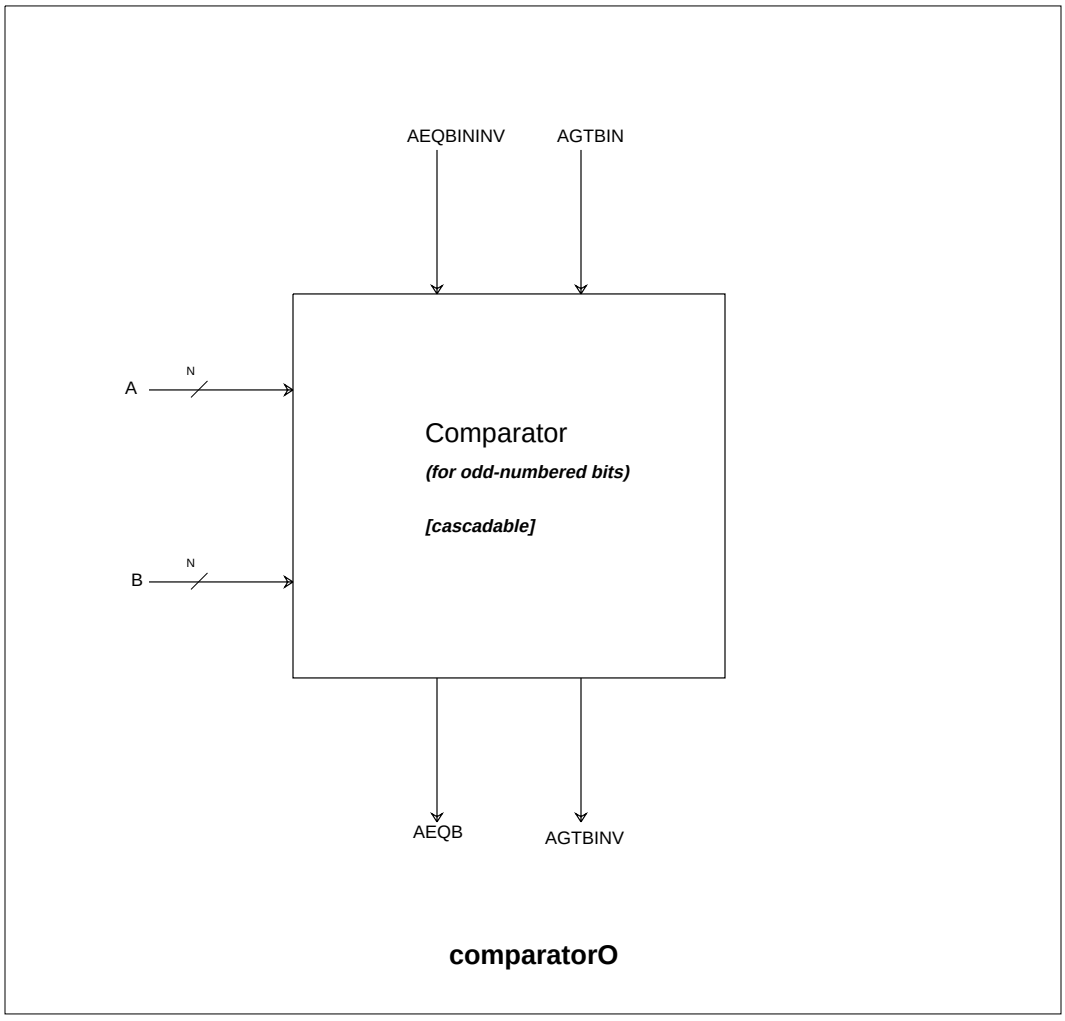
Description of the Terminals		
Terminal Name	Signal Type	Function
IN	N-bit input	input data
CNTOUT	output	status output
Vdd	power	connects to power supply
GND	ground	connects to ground

Thor model

```
if(fpack(IN,!N!-1,0)==!constant!) CNTOUT=ONE;
else if(fckbin(IN,!N!-1,0)==PASSED) CNTOUT=ZERO;
else fsetword(IN,!N!-1,0,UNDEF);
if(!N! > sizeof(int)*8) CNTOUT=UNDEF;
EXITMOD(0);
```

sdl File

```
(parent-cell compconst)
(parameters N constant)
(layout-generator TimLager)
(terminal IN (side LEFT RIGHT)(TERMTYPE DATA_SIGNAL)(DIRECTION INPUT))
(terminal CNTOUT (side TOP)(TERMTYPE CONTROL_SIGNAL)(DIRECTION OUTPUT))
(terminal Vdd (side TOP BOT)(TERMTYPE SUPPLY))
(terminal GND (side TOP BOT)(TERMTYPE GROUND))
(end-sdl)
```


AEQBININV
AGTBIN
A
N
B
N
Comparator
(for odd-numbered bits)
[cascadable]
AEQB
AGTBINV
comparatorO

```
(parameters N) ;works for odd number of bits only
(layout-generator TimLager)
(terminal A (side LEFT))
(terminal B (side LEFT))
(terminal AEQBININV (side TOP))
(terminal AGTBIN (side TOP))
(terminal AEQB (side BOT))
(terminal AGTBINV (side BOT))
(terminal GND (side BOT TOP ))
(terminal Vdd (side BOT TOP ))
(end-sdl)
```

Performance Results

Leafcell Delay Models				
Leafcell	From	To	Best Fit Linear Model	Root Mean Square Error
comparator_odd	AEQBIN	AEQBINV	$t_{plh} = 0.2517 + 3.9105 * C_{load}$	0.0105
			$t_{phl} = 0.3406 + 6.4677 * C_{load}$	0.0054
			$t_{lh} = 0.2549 + 9.0855 * C_{load}$	0.0819
			$t_{hl} = 0.3330 + 13.3800 * C_{load}$	0.0273
comparator_odd	AEQBIN	AGTB	$t_{plh} = 0.2517 + 3.9105 * C_{load}$	0.0105
			$t_{phl} = 0.3285 + 6.4722 * C_{load}$	0.0064
			$t_{lh} = 0.2549 + 9.0855 * C_{load}$	0.0819
			$t_{hl} = 0.3289 + 13.3816 * C_{load}$	0.0328
comparator_odd	B	AEQBINV	$t_{plh} = 1.1453 + 3.8802 * C_{load}$	0.0529
			$t_{phl} = 2.0859 + 6.4626 * C_{load}$	0.0406
			$t_{lh} = 0.3569 + 9.0810 * C_{load}$	0.1081
			$t_{hl} = 0.5315 + 13.2961 * C_{load}$	0.1234
comparator_odd	B	AGTB	$t_{plh} = 1.2084 + 3.8845 * C_{load}$	0.0669
			$t_{phl} = 1.5906 + 6.4287 * C_{load}$	0.0409
			$t_{lh} = 0.3462 + 9.0866 * C_{load}$	0.1081
			$t_{hl} = 0.3818 + 13.3592 * C_{load}$	0.0727
Note: delays are in ns, C_{load} is in pF				

6 comparator0

Synopsis

This block implements a comparator which can be used for datapaths with an odd number of bits. It takes two N-bit inputs A and B. The result of the comparison is indicated by the signals AEQB and AGTBINV which are enabled when the conditions A equal B and A greater than B respectively occur. The output AGTBINV is in negative logic. The inputs AEQBININV and AGTBIN should both usually be tied to 0. These inputs are actually meant for cascading purposes and may indicate the result of comparing the lower bits of A and B in another datapath. Similarly, the outputs AEQB and AGTBINV can be used as input to a comparator which is being used to compare the higher order bits of A and B in another datapath. It uses the leafcells *comparator_even* and *comparator_odd*.

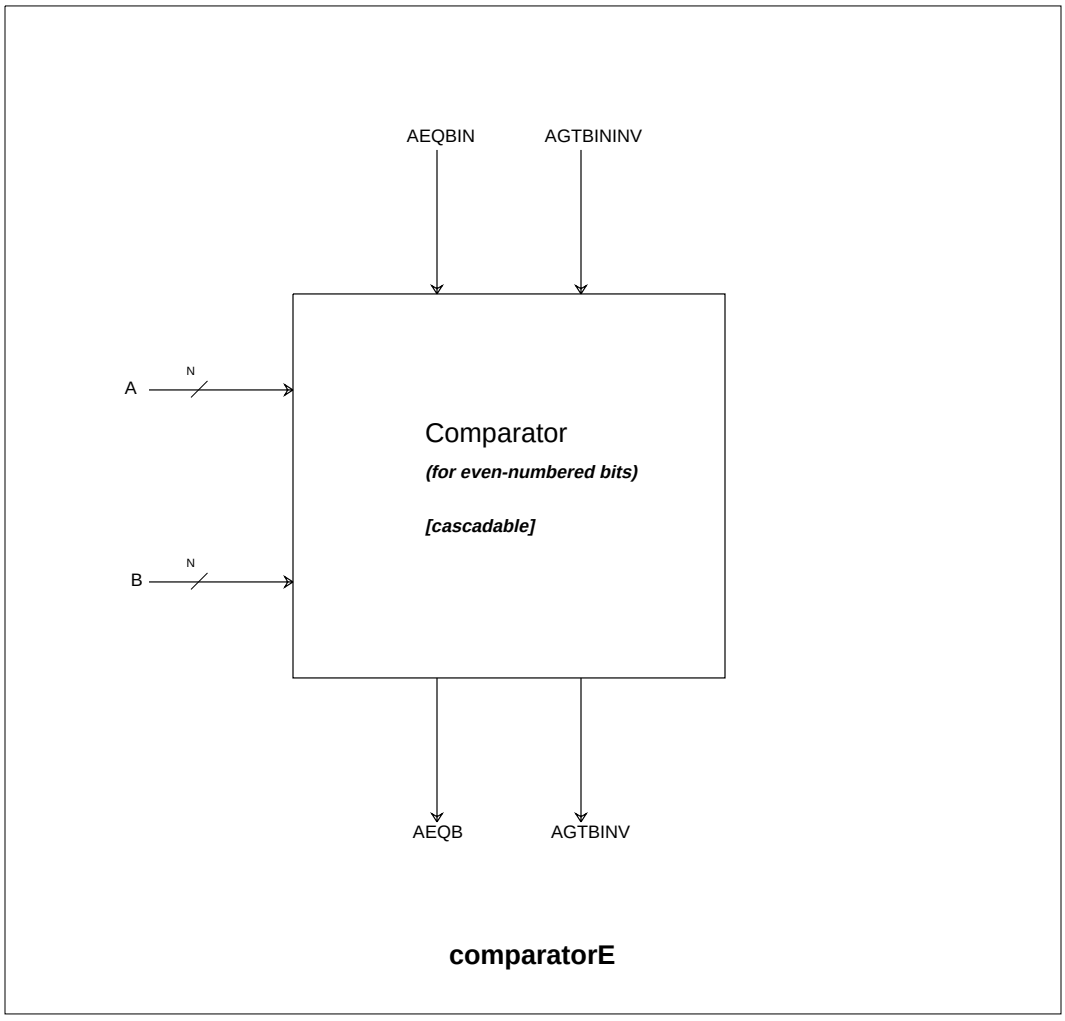
Description of the Terminals		
Terminal Name	Signal Type	Function
A	N-bit input	first operand
B	N-bit input	second operand
AEQB	output	high indicates A==B
AGTBINV	output	low indicates A>B
AEQBININV	input	usually tied to 0
AGTBIN	input	usually tied to 0
Vdd	power	connects to power supply
GND	ground	connects to ground

Behavioral Description

```
if ((AEQBININV==0) && (AGTBIN==1)) ERROR();
if (AEQBININV==0) {
    if (A==B) AEQB = AGTBINV = 1;
    else if (A>B) AEQB = AGTBINV = 0;
    else {AEQB =0; AGTBINV = 1;}
    }
else if (AGTBIN==1) {
    if (A>=B) AEQB = AGTBINV = 0;
    else {AEQB =0; AGTBINV = 1;}
    }
else {
    if (A>B) AEQB = AGTBINV = 0;
    else {AEQB =0; AGTBINV = 1;}
    }
```

sdl File

```
(parent-cell comparator0)
```


AEQBIN
AGTBININV
A
N
Comparator
(for even-numbered bits)
[cascadable]
B
N
AEQB
AGTBINV
comparatorE

```
(parameters N) ;works for even-numbered bits only
(layout-generator TimLager)
(terminal A (side LEFT))
(terminal B (side LEFT))
(terminal AEQBIN (side TOP))
(terminal AGTBININV (side TOP))
(terminal AEQB (side BOT))
(terminal AGTBINV (side BOT))
(terminal GND (side BOT TOP ))
(terminal Vdd (side BOT TOP ))
(end-sdl)
```

Performance Results

Leafcell Delay Models				
Leafcell	From	To	Best Fit Linear Model	Root Mean Square Error
comparator_even	AGTBIN	AEQB	$t_{plh} = 0.2986 + 6.6282 * C_{load}$	0.0157
			$t_{phl} = 0.3379 + 5.6045 * C_{load}$	0.0030
			$t_{lh} = 0.4027 + 15.5949 * C_{load}$	0.0399
			$t_{hl} = 0.2488 + 10.8109 * C_{load}$	0.0318
comparator_even	AGTBIN	AGTBINV	$t_{plh} = 0.2804 + 4.3410 * C_{load}$	0.0102
			$t_{phl} = 0.5179 + 5.6045 * C_{load}$	0.0030
			$t_{lh} = 0.3381 + 10.2628 * C_{load}$	0.0532
			$t_{hl} = 0.5142 + 10.8081 * C_{load}$	0.0382
comparator_even	B	AEQB	$t_{plh} = 0.9663 + 6.6206 * C_{load}$	0.0082
			$t_{phl} = 1.6424 + 5.6122 * C_{load}$	0.0490
			$t_{lh} = 0.4224 + 15.5806 * C_{load}$	0.0585
			$t_{hl} = 0.6069 + 10.7177 * C_{load}$	0.1472
comparator_even	B	AGTBINV	$t_{plh} = 1.9920 + 4.3542 * C_{load}$	0.0276
			$t_{phl} = 1.8698 + 5.6173 * C_{load}$	0.0169
			$t_{lh} = 0.3630 + 10.2483 * C_{load}$	0.0613
			$t_{hl} = 0.3293 + 10.7909 * C_{load}$	0.0594
Note: delays are in ns, C_{load} is in pF				

5 comparatorE

Synopsis

This block implements a comparator which can be used for datapaths with an even number of bits. It takes two N-bit inputs A and B. The result of the comparison is indicated by the signals AEQB and AGTBINV which are enabled when the conditions A equal B and A greater than B respectively occur. The output AGTBINV is in negative logic. The inputs AEQBIN and AGTBININV should both usually be tied to 1. These inputs are actually meant for cascading purposes and may indicate the result of comparing the lower bits of A and B in another datapath. Similarly, the outputs AEQB and AGTBINV can be used as input to a comparator which is being used to compare the higher order bits of A and B in another datapath. It uses the leafcells *comparator_even* and *comparator_odd*.

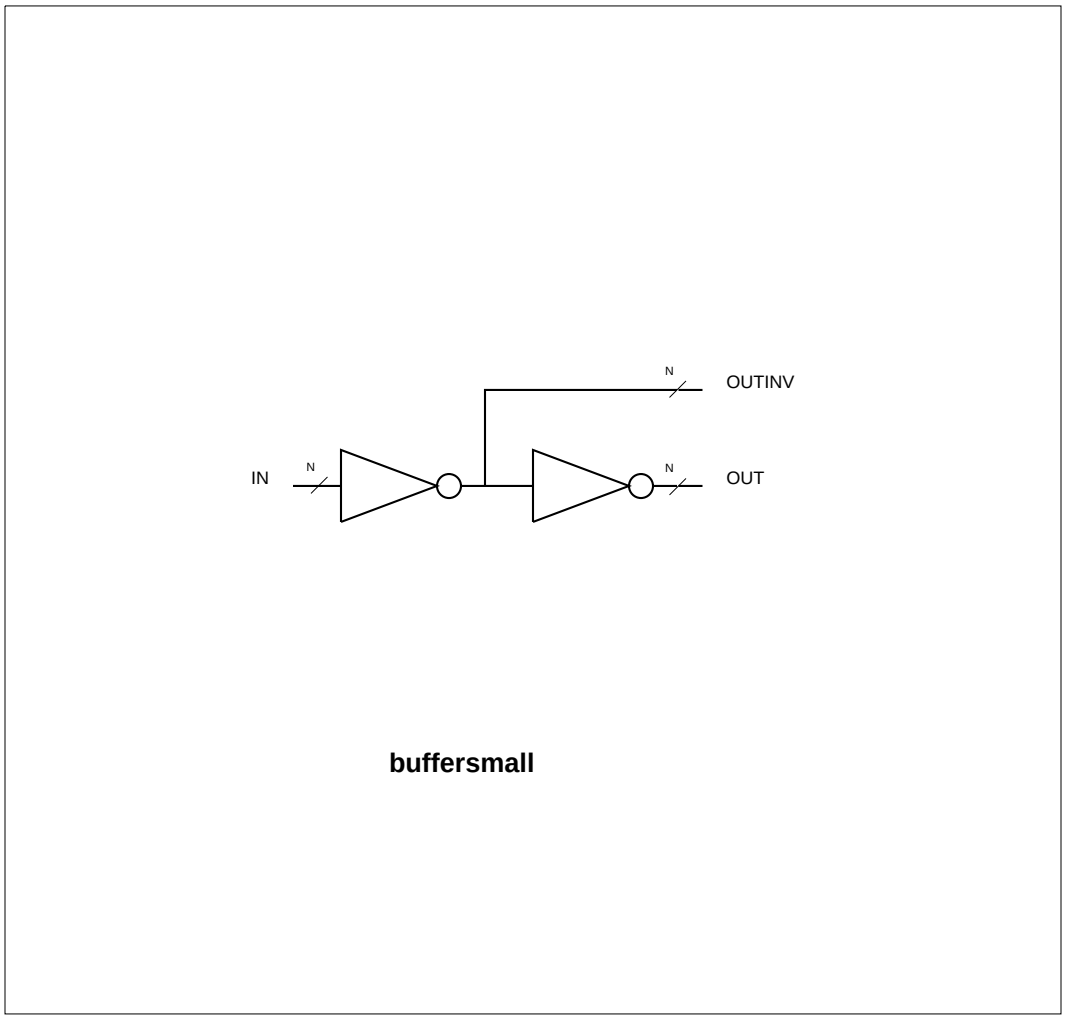
Description of the Terminals		
Terminal Name	Signal Type	Function
A	N-bit input	first operand
B	N-bit input	second operand
AEQB	output	high indicates A==B
AGTBINV	output	low indicates A>B
AEQBIN	input	usually tied to 1
AGTBININV	input	usually tied to 1
Vdd	power	connects to power supply
GND	ground	connects to ground

Behavioral Description

```
if ((AEQBIN==1) && (AGTBININV==0)) ERROR();
if (AEQBIN==1) {
    if (A==B) AEQB = AGTBINV = 1;
    else if (A>B) AEQB = AGTBINV = 0;
    else {AEQB =0; AGTBINV = 1;}
    }
else if (AGTBININV==0) {
    if (A>=B) AEQB = AGTBINV = 0;
    else {AEQB =0; AGTBINV = 1;}
    }
else {
    if (A>B) AEQB = AGTBINV = 0;
    else {AEQB =0; AGTBINV = 1;}
    }
```

sdl File

```
(parent-cell comparatorE)
```


N
OUTINV
IN
N
N
OUT
buffersmall

Performance Results

Leaf cell Delay Models				
Leaf cell	From	To	Best Fit Linear Model	Root Mean Square Error
buffersmall	IN	OUTINV	$t_{plh} = 1.8061 + 3.6843 * C_{load}$	1.7679
			$t_{phl} = 0.5980 + 5.4995 * C_{load}$	0.0606
			$t_{lh} = 0.9083 + 14.4212 * C_{load}$	0.1230
			$t_{hl} = 0.9009 + 10.8358 * C_{load}$	0.1458
buffersmall	IN	OUT	$t_{plh} = 1.2137 + 5.0647 * C_{load}$	0.9019
			$t_{phl} = 1.2310 + 2.5771 * C_{load}$	0.9326
			$t_{lh} = -1.0226 + 12.6611 * C_{load}$	3.4864
			$t_{hl} = 0.6488 + 5.4402 * C_{load}$	0.4930
Note: delays are in ns, C_{load} is in pF				

4 buffersmall

Synopsis

This block implements a buffer with small drive capability. The output is available in both positive and negative logic. It uses the leafcell *buffersmall*.

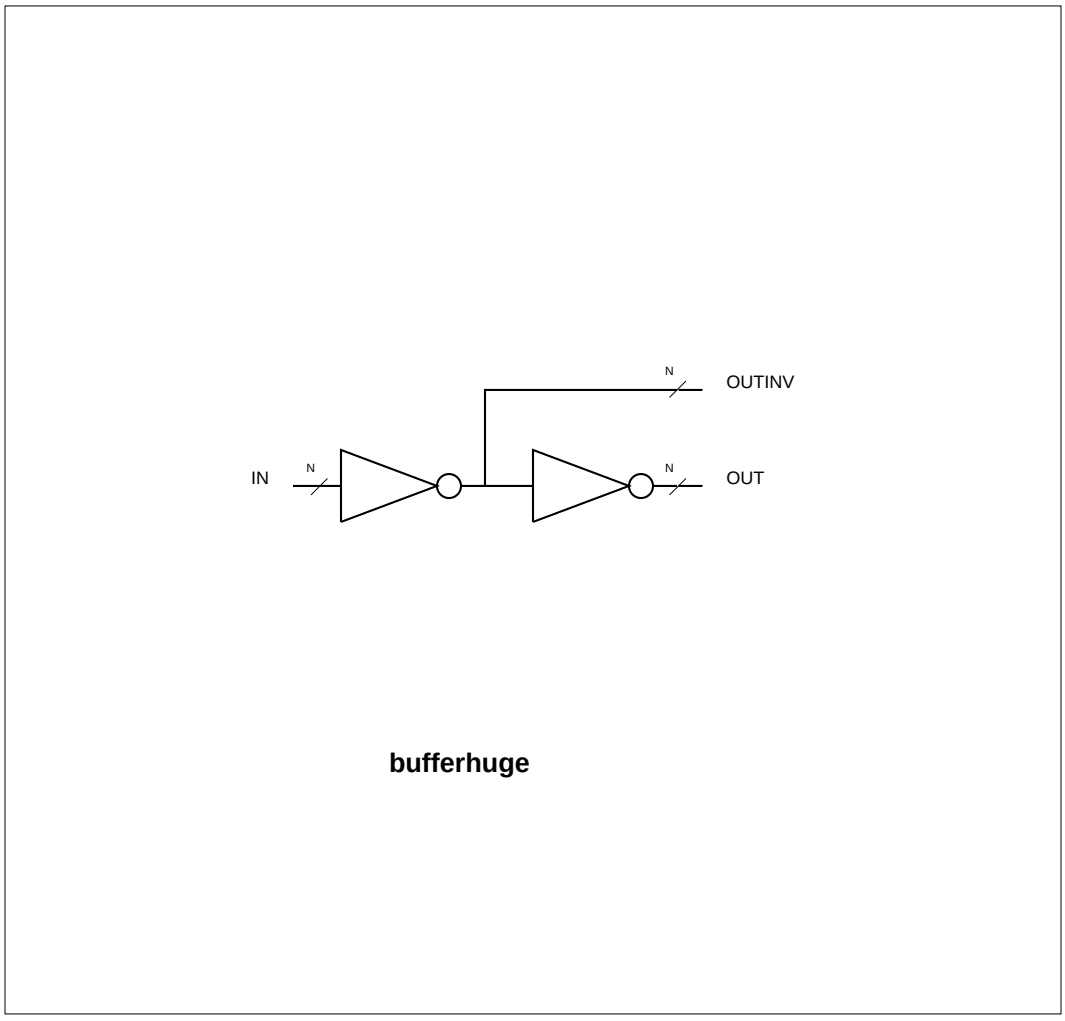
Description of the Terminals		
Terminal Name	Signal Type	Function
IN	N-bit input	input data
OUT	N-bit output	positive logic output
OUTINV	N-bit output	negative logic output
Vdd	power	connects to power supply
GND	ground	connects to ground

Behavioral Description

```
for (i=0;i<=N-1;i++) {
    OUT[i]=IN[i];
    OUTINV[i]=not(IN[i]);
    }
```

sdl File

```
(parent-cell buffersmall)
(parameters N)
(layout-generator TimLager)
(terminal GND (side TOP BOT ))
(terminal IN (side LEFT RIGHT ))
(terminal OUT (side LEFT RIGHT ))
(terminal OUTINV (side LEFT RIGHT ))
(terminal Vdd (side TOP BOT ))
(end-sdl)
```


N
OUTINV
IN
N
N
OUT
bufferhuge

Performance Results

Leafcell Delay Models				
Leafcell	From	To	Best Fit Linear Model	Root Mean Square Error
bufferhuge	IN	OUTINV	$t_{plh} = 0.6439 + 1.5157 * C_{load}$	0.0564
			$t_{phl} = 0.5573 + 1.3734 * C_{load}$	0.0077
			$t_{lh} = 1.2764 + 3.3596 * C_{load}$	0.0975
			$t_{hl} = 1.0548 + 2.7831 * C_{load}$	0.0417
bufferhuge	IN	OUT	$t_{plh} = 0.6676 + 0.6048 * C_{load}$	0.1257
			$t_{phl} = 0.6876 + 0.5804 * C_{load}$	0.1649
			$t_{lh} = 0.6942 + 1.3691 * C_{load}$	0.2862
			$t_{hl} = 0.6667 + 1.0300 * C_{load}$	0.3239
Note: delays are in ns, C_{load} is in pF				

3 bufferhuge

Synopsis

This block implements a buffer with large drive capability. The output is available in both positive and negative logic. It uses the leafcell *bufferhuge*.

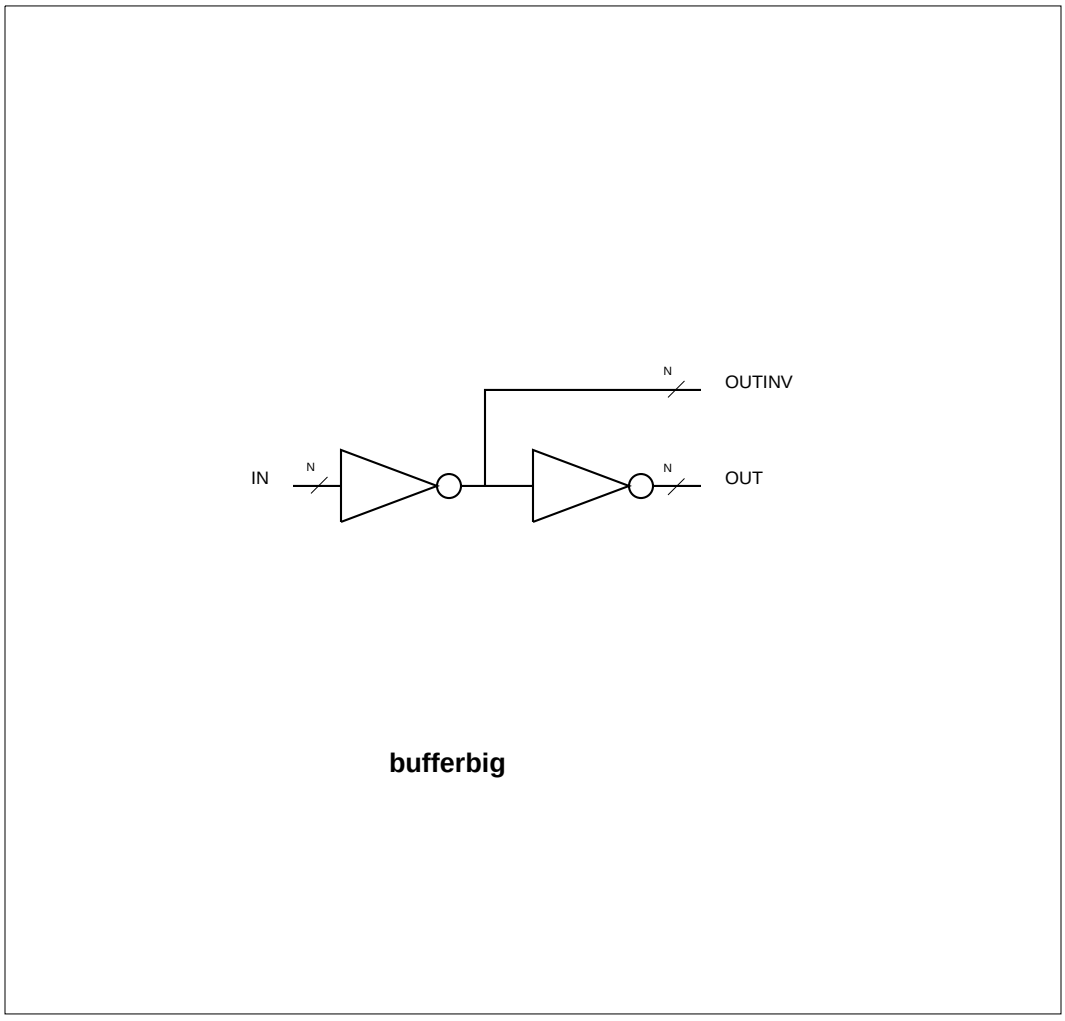
Description of the Terminals		
Terminal Name	Signal Type	Function
IN	N-bit input	input data
OUT	N-bit output	positive logic output
OUTINV	N-bit output	negative logic output
Vdd	power	connects to power supply
GND	ground	connects to ground

Behavioral Description

```
for (i=0;i<=N-1;i++) {
    OUT[i]=IN[i];
    OUTINV[i]=not(IN[i]);
    }
```

sdl File

```
(parent-cell bufferhuge)
(parameters N)
(layout-generator TimLager)
(terminal GND (side TOP BOT ))
(terminal IN (side LEFT RIGHT ))
(terminal OUT (side LEFT RIGHT ))
(terminal OUTINV (side LEFT RIGHT ))
(terminal Vdd (side TOP BOT ))
(end-sdl)
```


N
OUTINV
IN
N
N
OUT
bufferbig

Performance Results

Leaf cell Delay Models				
Leaf cell	From	To	Best Fit Linear Model	Root Mean Square Error
bufferbig	IN	OUTINV	$t_{plh} = 0.6868 + 3.9153 * C_{load}$	0.0331
			$t_{phl} = 0.3969 + 2.0567 * C_{load}$	0.0327
			$t_{lh} = 0.5909 + 9.2211 * C_{load}$	0.9481
			$t_{hl} = 0.7309 + 4.2211 * C_{load}$	0.2450
bufferbig	IN	OUT	$t_{plh} = 0.3423 + 3.0466 * C_{load}$	0.4472
			$t_{phl} = -0.0717 + 3.2748 * C_{load}$	1.2995
			$t_{lh} = 2.8267 + 1.5837 * C_{load}$	3.3804
			$t_{hl} = 0.9203 + 2.1680 * C_{load}$	0.4001
Note: delays are in ns, C_{load} is in pF				

2 bufferbig

Synopsis

This block implements a buffer with mediumdrive capability. The output is available in both positive and negative logic. It uses the leafcell *bufferbig*.

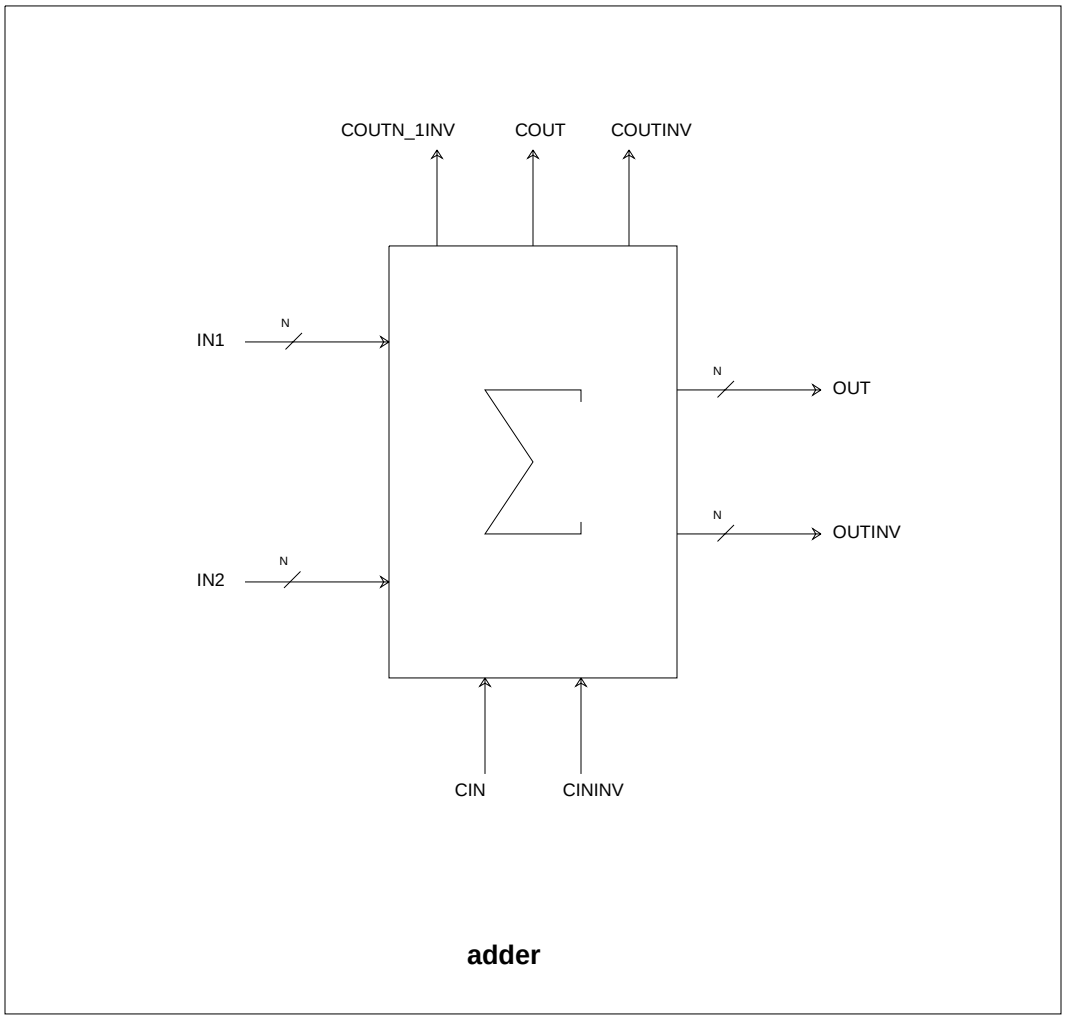
Description of the Terminals		
Terminal Name	Signal Type	Function
IN	N-bit input	input data
OUT	N-bit output	positive logic output
OUTINV	N-bit output	negative logic output
Vdd	power	connects to power supply
GND	ground	connects to ground

Behavioral Description

```
for (i=0;i<=N-1;i++) {
    OUT[i]=IN[i];
    OUTINV[i]=not(IN[i]);
    }
```

sdl File

```
(parent-cell bufferbig)
(parameters N)
(layout-generator TimLager)
(terminal GND (side TOP BOT ))
(terminal IN (side LEFT RIGHT ))
(terminal OUT (side RIGHT LEFT ))
(terminal OUTINV (side LEFT RIGHT ))
(terminal Vdd (side TOP BOT ))
(end-sdl)
```


COUTN_1INV
COUT
COUTINV
IN1
N
IN2
N
N
OUT
N
OUTINV
CIN
CININV
adder

Leafcell Delay Models				
Leafcell	From	To	Best Fit Linear Model	Root Mean Square Error
adder_odd	IN1	OUT	$t_{plh} = 1.8946 + 7.4565 * C_{load}$	0.1146
			$t_{phl} = 2.0700 + 7.3537 * C_{load}$	0.1174
			$t_{lh} = 2.2343 + 17.3421 * C_{load}$	0.1908
			$t_{hl} = 2.3803 + 15.6387 * C_{load}$	0.0941
		OUTINV	$t_{plh} = 2.6344 + 14.9128 * C_{load}$	0.2403
			$t_{phl} = 2.1499 + 14.8903 * C_{load}$	0.0519
			$t_{lh} = 1.7337 + 13.6427 * C_{load}$	0.0839
			$t_{hl} = 2.2807 + 11.7871 * C_{load}$	0.5075
	CIN	OUT	$t_{plh} = 1.2628 + 7.4656 * C_{load}$	0.1008
			$t_{phl} = 1.4055 + 7.3459 * C_{load}$	0.1390
			$t_{lh} = 2.6433 + 17.4017 * C_{load}$	0.2282
			$t_{hl} = 2.2934 + 15.6820 * C_{load}$	0.0550
		OUTINV	$t_{plh} = 1.9480 + 14.9141 * C_{load}$	0.2421
			$t_{phl} = 1.6621 + 14.8662 * C_{load}$	0.0412
			$t_{lh} = 1.7771 + 13.6224 * C_{load}$	0.0758
			$t_{hl} = 1.9704 + 12.0240 * C_{load}$	0.5764
adder_odd_nsb	IN1	OUT	$t_{plh} = 1.8793 + 7.4593 * C_{load}$	0.1216
			$t_{phl} = 2.0526 + 7.3536 * C_{load}$	0.1240
			$t_{lh} = 2.2195 + 17.3493 * C_{load}$	0.1816
			$t_{hl} = 2.3483 + 15.6432 * C_{load}$	0.0965
		OUTINV	$t_{plh} = 2.6398 + 14.9100 * C_{load}$	0.2334
			$t_{phl} = 2.1620 + 14.8859 * C_{load}$	0.0549
			$t_{lh} = 1.7509 + 13.6528 * C_{load}$	0.0728
			$t_{hl} = 2.3068 + 11.7860 * C_{load}$	0.5056
		COUT	$t_{plh} = 1.8108 + 9.1261 * C_{load}$	0.0939
			$t_{phl} = 2.4103 + 14.5070 * C_{load}$	0.1425
			$t_{lh} = 1.2975 + 11.3951 * C_{load}$	0.4781
			$t_{hl} = 1.9815 + 11.6961 * C_{load}$	0.0493
		COUTINV	$t_{plh} = 1.8846 + 7.2638 * C_{load}$	0.0272
			$t_{phl} = 1.2953 + 3.1729 * C_{load}$	0.0322
			$t_{lh} = 1.7281 + 16.8091 * C_{load}$	0.1949
			$t_{hl} = 1.0677 + 6.4218 * C_{load}$	0.5404
	CIN	OUT	$t_{plh} = 1.2528 + 7.4656 * C_{load}$	0.1008
			$t_{phl} = 1.3908 + 7.3431 * C_{load}$	0.1325
			$t_{lh} = 2.6346 + 17.4028 * C_{load}$	0.2127
			$t_{hl} = 2.2586 + 15.6892 * C_{load}$	0.0478
		OUTINV	$t_{plh} = 1.9534 + 14.9113 * C_{load}$	0.2351
			$t_{phl} = 1.6668 + 14.8690 * C_{load}$	0.0415
			$t_{lh} = 1.7922 + 13.6296 * C_{load}$	0.0859
			$t_{hl} = 2.0234 + 12.0113 * C_{load}$	0.5565
		COUT	$t_{plh} = 0.9915 + 9.1595 * C_{load}$	0.2533
			$t_{phl} = 1.0707 + 10.7091 * C_{load}$	0.2134
			$t_{lh} = 1.0780 + 11.4361 * C_{load}$	0.4226
			$t_{hl} = 1.1717 + 10.1788 * C_{load}$	0.3139
		COUTINV	$t_{plh} = 0.5904 + 4.3704 * C_{load}$	0.0223
			$t_{phl} = 0.4931 + 3.1823 * C_{load}$	0.0081
			$t_{lh} = 1.1867 + 10.1667 * C_{load}$	0.0245
			$t_{hl} = 0.8616 + 6.5082 * C_{load}$	0.1295
Note: delays are in ns, C_{load} is in pF				

Performance Results

Leafcell Delay Models				
Leafcell	From	To	Best Fit Linear Model	Root Mean Square Error
adder_even	IN1	OUT	$t_{plh} = 2.6625 + 12.9976 * C_{load}$	0.1923
			$t_{phl} = 2.9878 + 17.1631 * C_{load}$	0.0854
			$t_{lh} = 2.5146 + 11.5248 * C_{load}$	0.4324
			$t_{hl} = 2.3779 + 12.9045 * C_{load}$	0.4285
		OUTINV	$t_{plh} = 2.6746 + 9.1028 * C_{load}$	0.1505
			$t_{phl} = 2.2989 + 6.4742 * C_{load}$	0.0433
			$t_{lh} = 3.1548 + 21.3831 * C_{load}$	0.1831
			$t_{hl} = 2.4453 + 13.1485 * C_{load}$	0.3565
	CIN	OUT	$t_{plh} = 1.8136 + 12.9234 * C_{load}$	0.1915
			$t_{phl} = 2.0971 + 17.1004 * C_{load}$	0.0433
			$t_{lh} = 1.7146 + 11.8858 * C_{load}$	0.8349
			$t_{hl} = 2.0712 + 12.9988 * C_{load}$	0.1897
		OUTINV	$t_{plh} = 1.5774 + 9.1391 * C_{load}$	0.0512
			$t_{phl} = 1.2830 + 6.5020 * C_{load}$	0.0499
			$t_{lh} = 2.8238 + 21.5207 * C_{load}$	0.0817
			$t_{hl} = 2.1112 + 13.2208 * C_{load}$	0.2388
adder_even_msb	IN1	OUT	$t_{plh} = 2.7668 + 13.0007 * C_{load}$	0.1819
			$t_{phl} = 3.0729 + 17.0460 * C_{load}$	0.0634
			$t_{lh} = 2.5799 + 11.5293 * C_{load}$	0.4066
			$t_{hl} = 2.4902 + 12.8290 * C_{load}$	0.3824
		OUTINV	$t_{plh} = 2.7868 + 8.9860 * C_{load}$	0.1088
			$t_{phl} = 2.3869 + 6.4714 * C_{load}$	0.0369
			$t_{lh} = 3.3600 + 21.1073 * C_{load}$	0.2133
			$t_{hl} = 2.5478 + 13.1558 * C_{load}$	0.4240
		COUT	$t_{plh} = 1.3401 + 4.3633 * C_{load}$	0.0940
			$t_{phl} = 1.7074 + 4.1684 * C_{load}$	0.0600
			$t_{lh} = 1.1152 + 10.2315 * C_{load}$	0.3977
			$t_{hl} = 1.6685 + 7.9039 * C_{load}$	0.3837
		COUTINV	$t_{plh} = 2.3381 + 9.7628 * C_{load}$	0.2392
			$t_{phl} = 1.8869 + 10.7250 * C_{load}$	0.0721
			$t_{lh} = 1.9651 + 10.1902 * C_{load}$	0.4868
			$t_{hl} = 1.4990 + 10.0766 * C_{load}$	0.4955
	CIN	OUT	$t_{plh} = 1.9094 + 12.9249 * C_{load}$	0.1834
			$t_{phl} = 2.2105 + 16.9654 * C_{load}$	0.0542
			$t_{lh} = 1.9318 + 11.7958 * C_{load}$	0.7092
			$t_{hl} = 2.1266 + 12.9497 * C_{load}$	0.2203
		OUTINV	$t_{plh} = 1.6449 + 9.0392 * C_{load}$	0.0428
			$t_{phl} = 1.3597 + 6.5003 * C_{load}$	0.0583
			$t_{lh} = 2.9490 + 21.3156 * C_{load}$	0.1277
			$t_{hl} = 2.1692 + 13.2423 * C_{load}$	0.2529
		COUT	$t_{plh} = 0.6666 + 4.3277 * C_{load}$	0.0661
			$t_{phl} = 0.5234 + 3.1820 * C_{load}$	0.0288
			$t_{lh} = 1.1564 + 10.1986 * C_{load}$	0.1245
			$t_{hl} = 1.0923 + 6.5002 * C_{load}$	0.1740
		COUTINV	$t_{plh} = 0.9259 + 8.5796 * C_{load}$	0.1514
			$t_{phl} = 1.0516 + 10.7838 * C_{load}$	0.1031
			$t_{lh} = 1.2127 + 10.2193 * C_{load}$	0.5469
			$t_{hl} = 1.6146 + 10.0858 * C_{load}$	0.6915
Note: delays are in ns, C_{load} is in pF				

sdl File

```
(parent-cell adder)
(parameters N)
(layout-generator TimLager)
(terminal CIN (side BOT))
(terminal CININV (side BOT))
(terminal COUT (side TOP))
(terminal COUTINV (side TOP))
(terminal COUTN_1INV (side TOP))
(terminal GND (side TOP BOT ))
(terminal IN1 (side LEFT))
(terminal IN2 (side LEFT))
(terminal OUT (side RIGHT))
(terminal OUTINV (side LEFT))
(terminal Vdd (side TOP BOT ))
(end-sdl)
```

1 adder

Synopsis

This N-bit ripple carry adder, uses a dual carry chain for fast carry propagation. Supplied with two N-bit inputs (IN1 and IN2), a single carry input (CIN), and a complemented carry input (CININV – the logical inverse of CIN), the adder produces an N-bit sum, which is available in both positive and negative logic as OUT and OUTINV, respectively. The carry output from the most significant bit (msb) is also available in true and form as COUT and COUTINV, respectively. Finally, the carry output of the penultimate bit (the one before the msb) is available in negative logic as COUTN_1INV; its inverse can be xor-ed with COUT to check for overflow or underflow. The least significant N-1 bits of the adder use the leafcells *adder_even* and *adder_odd* in the even and odd bit positions, and the most significant bit uses either *adder_even_msb* or *adder_odd_msb* depending on whether the msb bit position is even or odd.

Description of the Terminals		
Terminal Name	Signal Type	Function
IN1	N-bit input	first operand
IN2	N-bit input	second operand
OUT	N-bit output	IN1 plus IN2
OUTINV	N-bit output	not(IN1 plus IN2)
COUT	output	carry out of msb
COUTINV	output	not(carry out of msb)
COUTN_1INV	output	not(carry into msb)
CIN	input	carry-in into lsb
CININV	input	not(carry-in) into lsb
Vdd	power	connects to power supply
GND	ground	connects to ground

Behavioral Description

```
if (CIN==CININV) ERROR();
CARRY[0]=CIN;
for (i=0;i<=N-1;i++) {
    OUT[i]=xor(xor(IN1[i],IN2[i]),CARRY[i]);
    OUTINV[i]=not(OUT[i]);
    CARRY[i+1]=or(and(IN1[i],IN2[i]),and(CARRY[i],or(IN1[i],IN2[i])));
    }
COUT=CARRY[N];
COUTINV=not(CARRY[N]);
COUTN_1INV=not(CARRY[N-1]);
```

Contents

LagerIV Datapath Block Library Documentation

University of California, Berkeley

December 3, 1988

This report documents the datapath blocks available for designing bit-slice datapaths in the LagerIV IC design environment.