**High-Level Model of c7552**

**Statistics:** 207 inputs; 108 outputs; 3512 gates

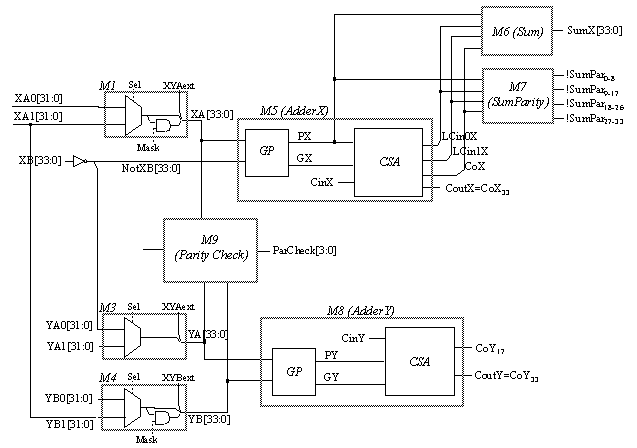
**Function:** 34-bit adder and magnitude comparator with input parity checking

This benchmark circuit contains a 34-bit adder (M5), a 34-bit magnitude comparator (M8) using another 34-bit adder, and a parity checker (M9). Each of the XA, YA, and YB buses is fed by a set of 2:1 multiplexers controlled by the Sel input. Bits 31-22 of XA and YB can be set to logic 0 with the Mask input. The two adders M5 and M8 are identical, and are of carry select type, as are those of c5315. They consist of alternating 4- and 5-bit blocks, with the last block being 2 bits. The comparator (M8) of this benchmark is similar to that of c2670. It performs the comparison YB>XB (if Sel=0) or YB>!YA1 (if Sel=1) by calculating YB+!XB (if Sel=0) or YB+!YA1 (if Sel=1) (Note: the input bus YA1 is assumed to be inverted). The comparator has an output (CoutY) for the whole 34-bit inputs as well as an output (CoutY\_17) for the 17-bit portion of its inputs. Module M7 calculates the parity for the following four parts of the adder output SumX: SumX[8:0], SumX[17:9], SumX[26:18], SumX[33:27]. Module M9 appears to be a type of sanity checker that calculates the AND of the parities of all its inputs.

[**Inputs/Outputs vs. Netlist numbers**](http://web.eecs.umich.edu/%7Ejhayes/iscas.restore/c7552/c7552.html#pgfId=1015521)

**Models:**

* I. Original ISCAS gate-level netlist
  + [in ISCAS-89 format](http://web.eecs.umich.edu/%7Ejhayes/iscas.restore/c7552/c7552.isc)
  + [in Verilog](http://web.eecs.umich.edu/%7Ejhayes/iscas.restore/c7552/c7552gate.v)
* II. [Verilog hierarchical netlist](http://web.eecs.umich.edu/%7Ejhayes/iscas.restore/c7552/c7552high.v) (functionally equivalent to I)
* III. [Verilog flat netlist](http://web.eecs.umich.edu/%7Ejhayes/iscas.restore/c7552/flat7552.v) (flat version of II; functionally equivalent to I, but with minor structural differences)



**Detailed bus definitions:**

- XA:

* XA[21:0] = XA0[21:0] if Sel=0,   XA1[21:0] if Sel=1
* XA[31:22] = XA0[21:0].Mask if Sel=0,   XA1[21:0].Mask if Sel=1
* XA[32] = XA[33] = XYAext

- NotXB:

* NotXB[0] = ! XB[0] if Sel=1,   **logic 0** if Sel=0
* NotXB[31:1] = ! XB[31:1]
* NotXB[33:32] = ! ( XB[33:32]. XYBext )

- YA:

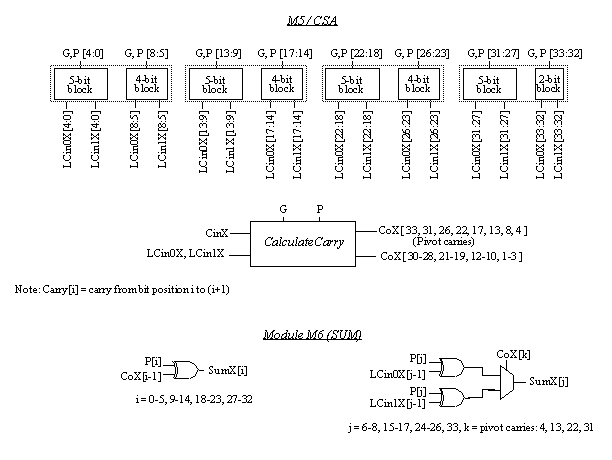
* YA[0] = **logic 1** if Sel=0,   YA1[0] if Sel=1
* YA[31:1] = NotXB[31:1] if Sel=0,   YA1[31:1] if Sel=1
* YA[32] = YA[33] = XYAext.

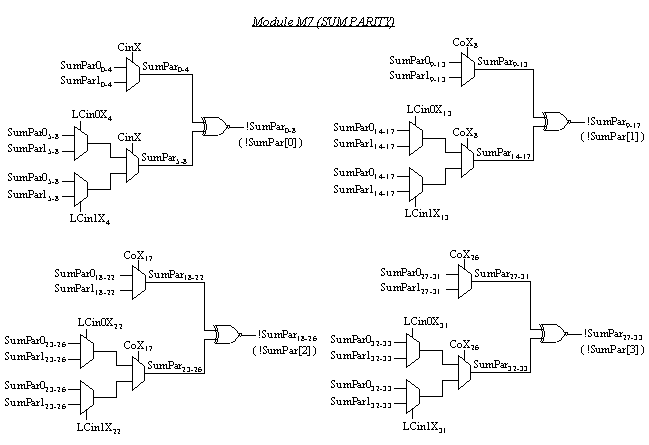
- YB:

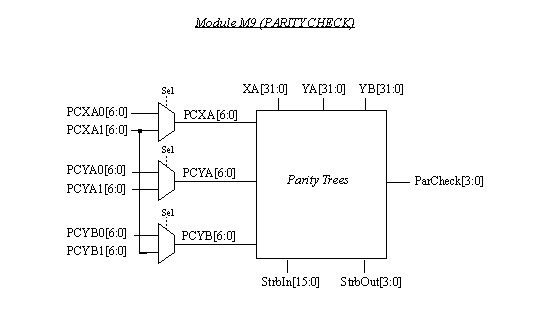
* YB[21:0] = YB0[21:0] if Sel=0,   XA1[21:0] if Sel=1
* YB[31:22] = YB0[21:0].Mask if Sel=0,   XA1[21:0].Mask if Sel=1
* YB[33:32] = YB0[33:32] + ! XYBext

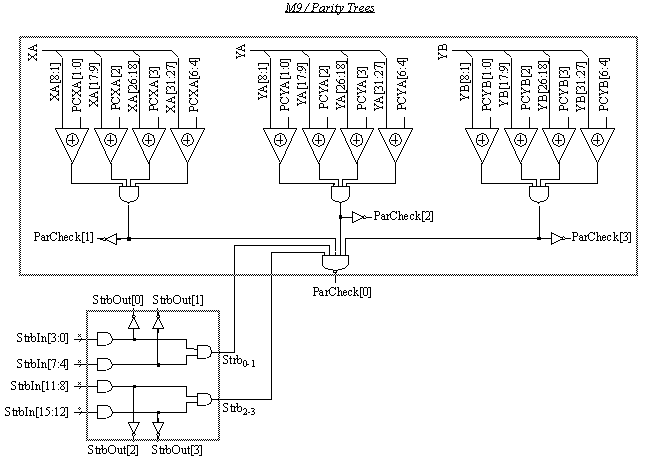
- XBbuf[33:0] = XB[33:0]

- PCYA0buf[3:0] = { PCYA0[6], PCYA0[3], PCYA0[2], PCYA0[0] }









|  |  |
| --- | --- |
| Input | Netlist numbers |
| XA0[31:0] | 213, 214, 215, 216, 209, 153, 154, 155, 156, 157, 158, 159, 160, 151, 219, 220, 221, 222, 223, 224, 225, 226, 217, 231, 232, 233, 234, 235, 236, 237, 238, **logic 0** |
| XA1[31:0] | 10 \*{**logic 1**}, 135, 144, 138, 147, 66, 50, 32, 35, 47, 121, 94, 97, 118, 100, 124, 127, 130, 103, 23, 26, 29, 41 |
| XB[33:0] | 1496, 1492, 1486, 1480, 106, 1469, 1462, 2256, 2253, 2247, 2239, 2236, 2230, 2224, 2218, 2211, 4437, 4432, 4427, 4420, 4415, 4410, 4405, 4400, 4394, 3749, 3743, 3737, 3729, 3723, 3717, 3711, 3705, 3701 |
| YA1[31:0] | 88, 112, 87, 111, 113, 110, 109, 86, 63, 64, 85, 84, 83, 65, 62, 61, 60, 79, 80, 81, 59, 78, 77, 56, 55, 54, 53, 73, 75, 76, 74, 70 |
| YB0[33:0] | 2204, 1455, 166, 167, 168, 169, **logic 1**, 173, 174, 175, 176, 177, 178, 179, 180, 171, 189, 190, 191, 192, 193, 194, 195, 196, 187, 200, 201, 202, 203, 204, 205, 206, 207, **logic 0** |
| !Sel | 18 |
| CinX, CinY | 4526, 89 |
| Mask=!Mask1+!Mask2  Mask1, Mask2 | 112, 9 |
| XYAext, XYBext | 38, 4528 |
| PCXA0[6:0] | **logic 1**, 211, 212, 161, 227, 239, 229 |
| PCXA1[6:0] | 3\*{**logic 1**}, 141, 115, 44, 41 |
| PCYA0[6:0] | 1459, 1496, 1492, 2208, 4398, 3701, 3698 |
| PCYA1[6:0] | 114, 2204, 1455, 82, 58, 70, 69 |
| PCYB0[6:0] | 170, 164, 165, 181, 197, 208, 198 |
| StrbIn[15:0] | 199, 188, 172, 162, 186, 185, 182, 183, 230, 218, 152, 210, 240, 228, 184, 150 |
| MiscIn[7:0] | 57, 5, 133, 134, 1197, 15, 163, 1 |

|  |  |
| --- | --- |
| Output | Netlist numbers |
| SumX[33:0] | 469, 471, 327, 330, 333, 336, 324, 298, 301, 304, 307, 310, 313, 316, 319, 295, 347, 350, 353, 356, 359, 362, 365, 368, 344, 376, 379, 382, 385, 388, 391, 394, 397, 373 |
| !SumPar[3:0] | 338, 321, 370, 399 |
| CoutX1, CoutX2 | (270, 246) [\*](http://web.eecs.umich.edu/%7Ejhayes/iscas.restore/c7552/c7552.html#pgfId=1015519) , (273, 276) [\*](http://web.eecs.umich.edu/%7Ejhayes/iscas.restore/c7552/c7552.html#pgfId=1015519) |
| CoutY1, CoutY2 | (258, 264) [\*](http://web.eecs.umich.edu/%7Ejhayes/iscas.restore/c7552/c7552.html#pgfId=1015519) , 249 |
| CoutY\_17 | 252 |
| ParCheck[3:0] | 416, 414, 412, 418 |
| XBbuf[33:0] | 440, 438, 442, 444, 446, 448, 436, 480, 482, 484, 486, 488, 490, 492, 494, 478, 524, 526, 528, 530, 532, 534, 536, 538, 522, 544, 546, 548, 550, 552, 554, 556, 558, 542 |
| StrbOut | 410, 408. 406, 404 |
| PCYA0buf[3:0] | 450, 496, 540, 560 |
| MiscOut[5:0] | 402, 289, 292, 279, 278, 2 |

\* (a,b): a,b are identical outputs.