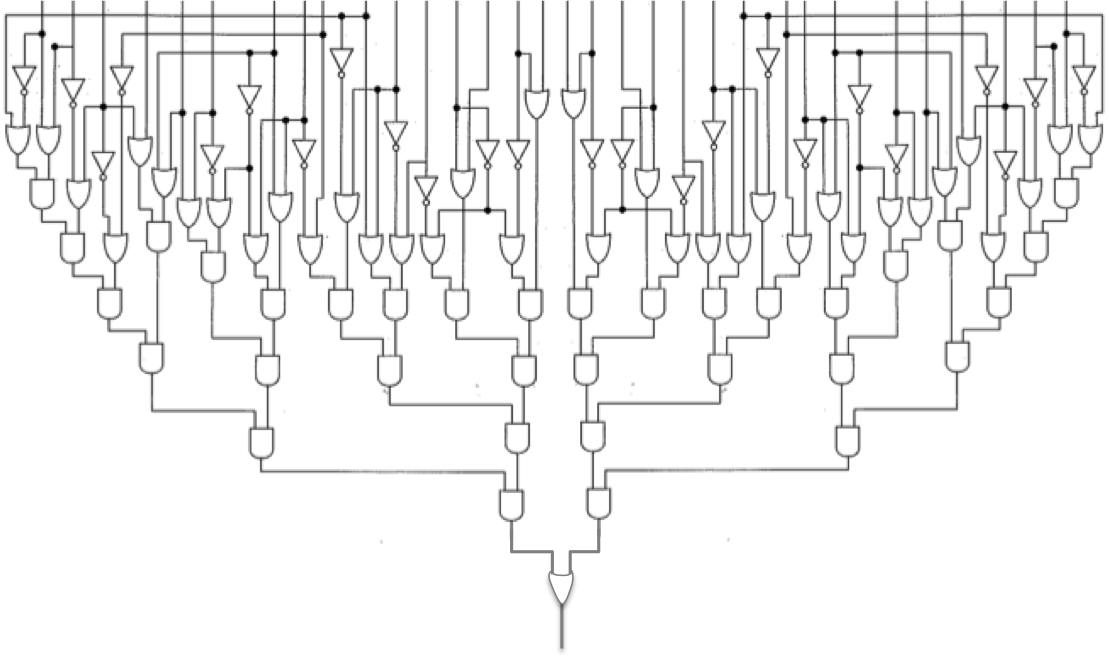
**(Using tree-structured communication 50 points or using serial communication 40 points)** The Circuit Satisfiability Problem for a given binary circuit is to find the set of inputs that cause that circuit to produce '1' as its output. For example, here is a 32-bit circuit diagram:



Download the program circuitSatisfiability.c.

1. Using your favorite text editor, modify circuitSatisfiability.c so that it uses the MPI\_Wtime() function to time the computation:

...

double startTime = 0.0, totalTime = 0.0;

startTime = MPI\_Wtime();

for (i = 0; i < UINT\_MAX; i++) {

count += checkCircuit(id, i);

}

totalTime = MPI\_Wtime() - startTime;

printf("Process %d finished in time %f secs.\n", id, totalTime);

...

With these modifications, the program will self-report how long it took to check the circuit.

1. Use MPI's version of the reduction pattern to sum the distributed processes' count-values into a global count, and have process 0 output this global count.
2. When your program compiles correctly and produces the correct results, time its execution with 1, 2, 4, 8, and 16 processes. Create a line-chart of your execution times, with the number of processes on the X-axis and time on the Y-axis. Label the scale of your X-axis with the number of processes (1, 2, 4, 8, and 16) so that this axis indicates the scale at which measurements were taken.
3. **(Using tree-structured communication 50 points or using serial communication 40 points)**





**Please upload your report and source code to Moodle.**