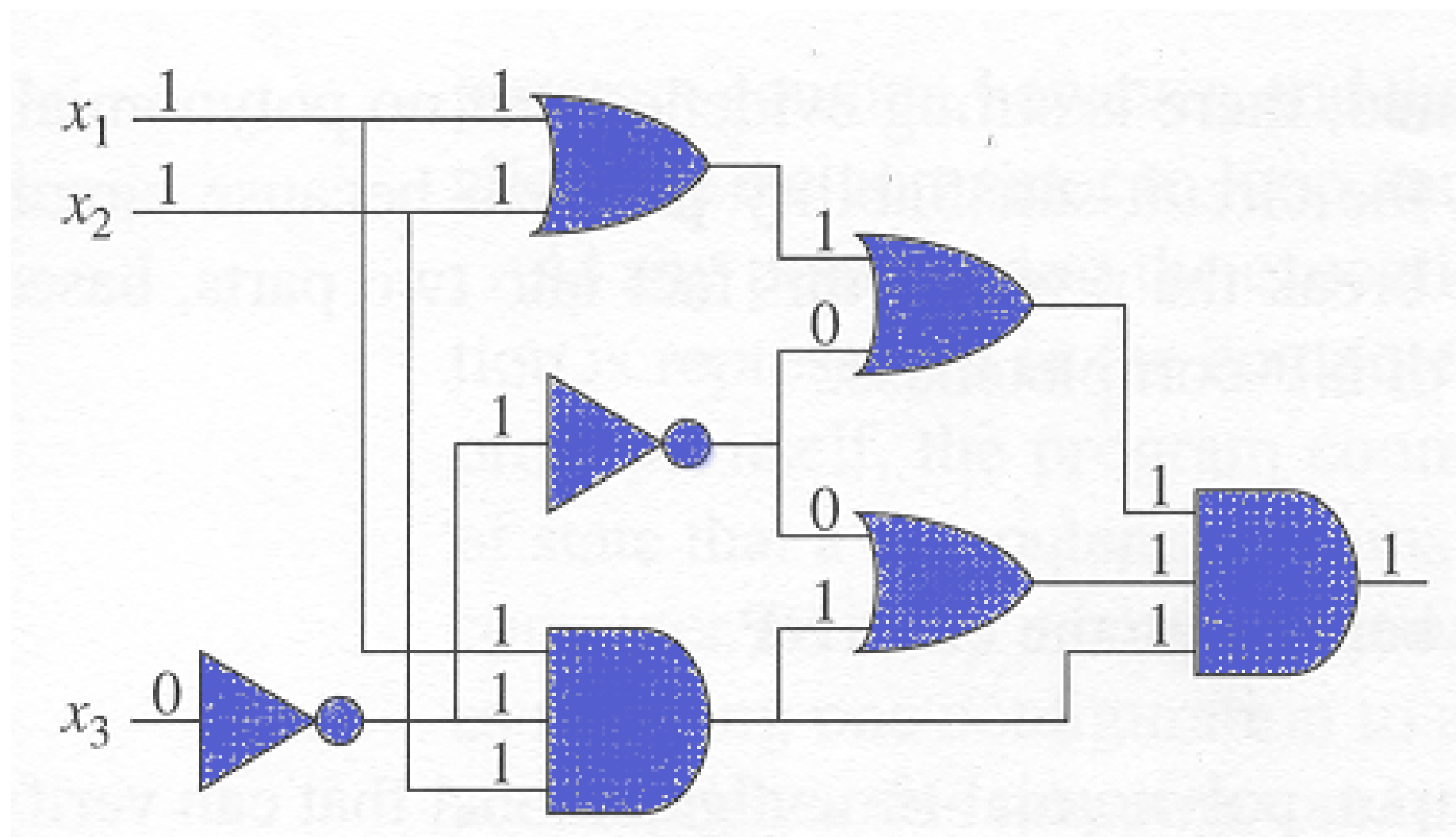


# Circuit Satisfiability Example

## Configuration



This logical formula  $y = f(x_1, x_2, x_3)$  is **satisfied** by input values such as (TRUE,TRUE,FALSE) which produce a result of TRUE.

## Model Problem

Our model formula uses 31 clauses in 23 variables, and begins:

$$y = (x_1 \vee x_2) \wedge (-x_2 \vee -x_4) \wedge (x_3 \vee x_4) \wedge \dots$$

There are  $2^{23}$  different choices for X.

Only 15 of these result in a value of TRUE.

Finding all inputs with a TRUE result is the **circuit satisfiability problem**.

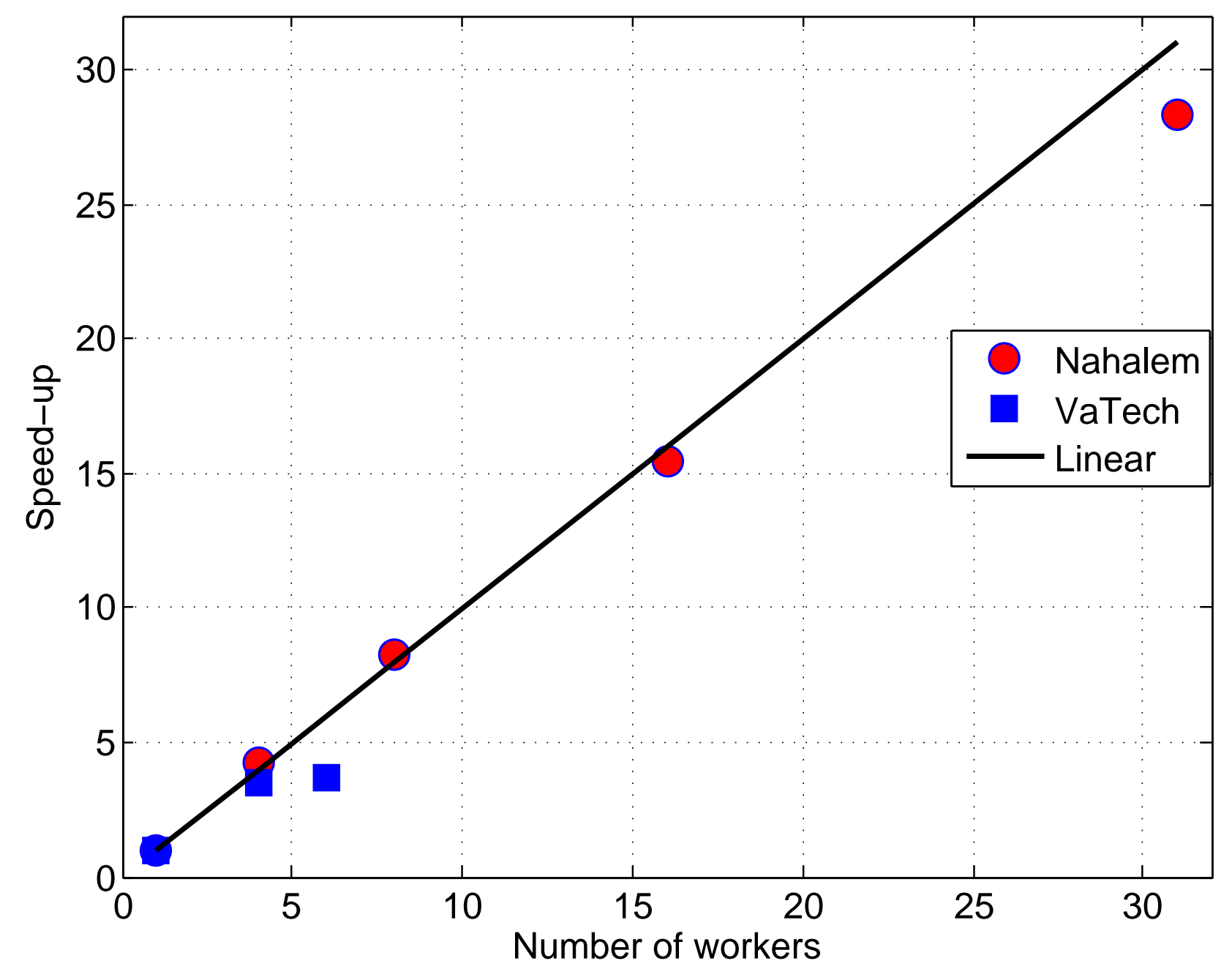
Our solution is an exhaustive search: generate and test every possible input.

If multiple workers are available, they can search in parallel.

## Code Fragment

```
n = 23;
sol_num = 0;
parfor i = 0 : 2^n - 1
    % Binary I yields logical X;
    x = i4_to_bvec ( i, n );
    value = circuit_value ( n, x );
    if ( value == 1 )
        sol_num = sol_num + 1;
        print_bvec ( x );
    end
end
```

## Timing



## Discussion

The small Virginia Tech cluster got the expected parallel improvement with 4 workers, but could not take much advantage of 8. The Nehalem cluster showed a near perfect parallel improvement across the range of workers investigated. This problem is an ideal case for parallel processing; there is no interaction between workers except to report the solutions.

## Reference

[http://people.sc.fsu.edu/~burkardt/...](http://people.sc.fsu.edu/~burkardt/)

- Burkardt, Cliff, Snow,  
*MATLAB Parallel Programming:  
Timing Results on an Intel Nehalem Cluster,*  
...pdf/nehalem\_matlab.pdf.
- ...m\_src/md\_parallel/md\_parallel.html.
- .../m\_src/satisfiability\_parallel  
/satisfiability\_parallel.html.