The GSAT Chip

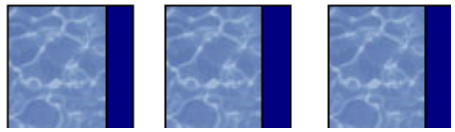
The SAT Problem

Given a Boolean expression of the form:

$$(A+\bar{B}+C)\cdot(\bar{D}+E+\bar{A})\cdot(B+\bar{C}+\bar{E})\ldots$$

Choose truth values for the variables such that the expression is satisfied.

Is this hard?
Computer science says yes: it's NP- Complete.



Is This Useful?

- This form (3CNF) is completely general
- Satisfiability finds application in:
  - Project scheduling (Continental and NASA even use the GSAT algorithm)
  - Instruction scheduling for compilers
  - Logic network verification

The GSAT Algorithm

A hill -climbing incremental improvement technique

- Randomly initialize the variables
- Until all of the clauses are satisfied:
  - Select a variable *ai* at random
  - Complement *ai*
  - If that decreased the number of satisfied clauses, undo it
- Output *a1*…*an*

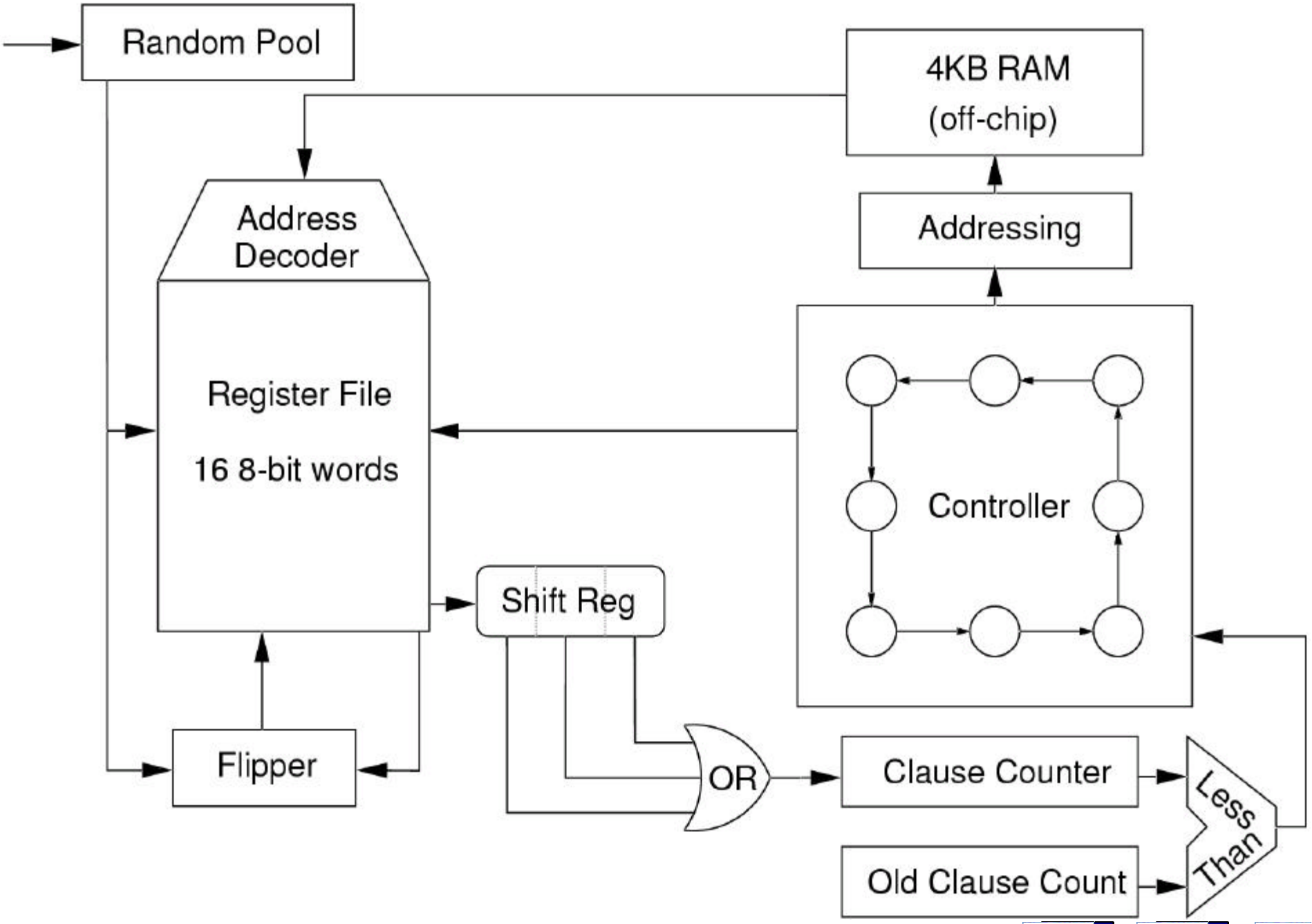
Chip Capabilities

- Problems involving up to 128 variables
- Expressions with as many as 1500 clauses
- Pause at any time and have random access to the current variables

Major Components

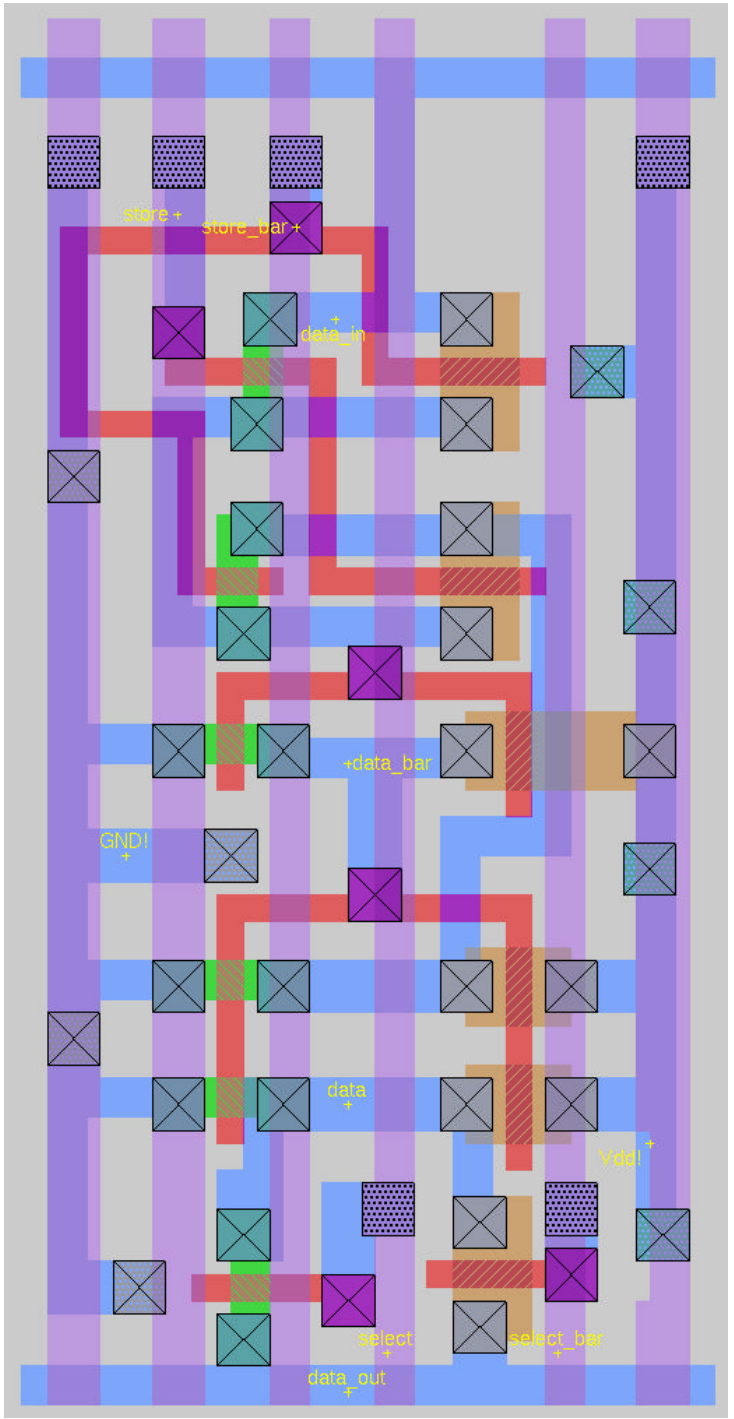
- 128 bits of on-chip register space
- 4 KB address space for off-chip memory
- Counters for memory addressing and expression evaluation
- A controller to manage it all

System Block Diagram

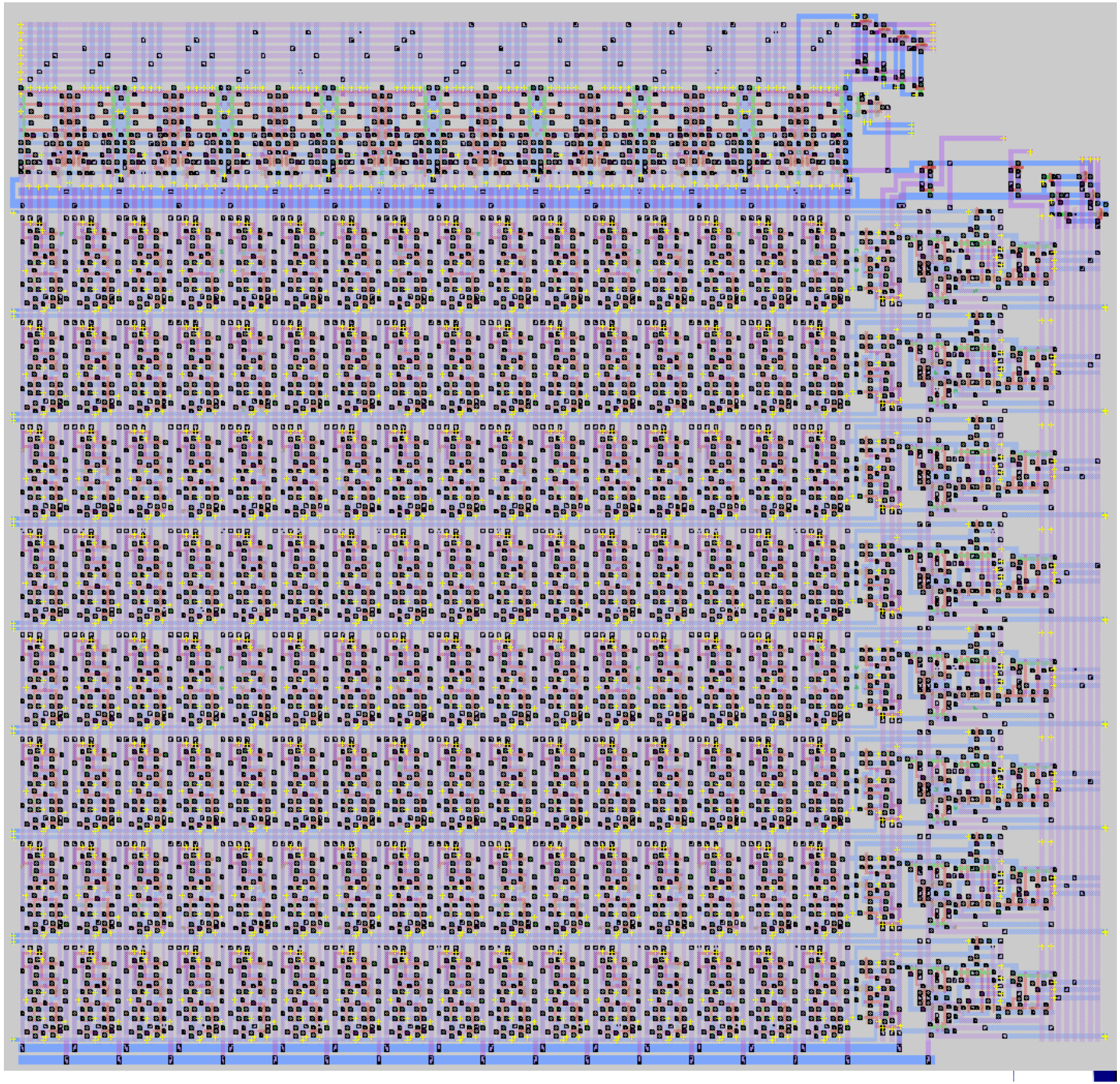


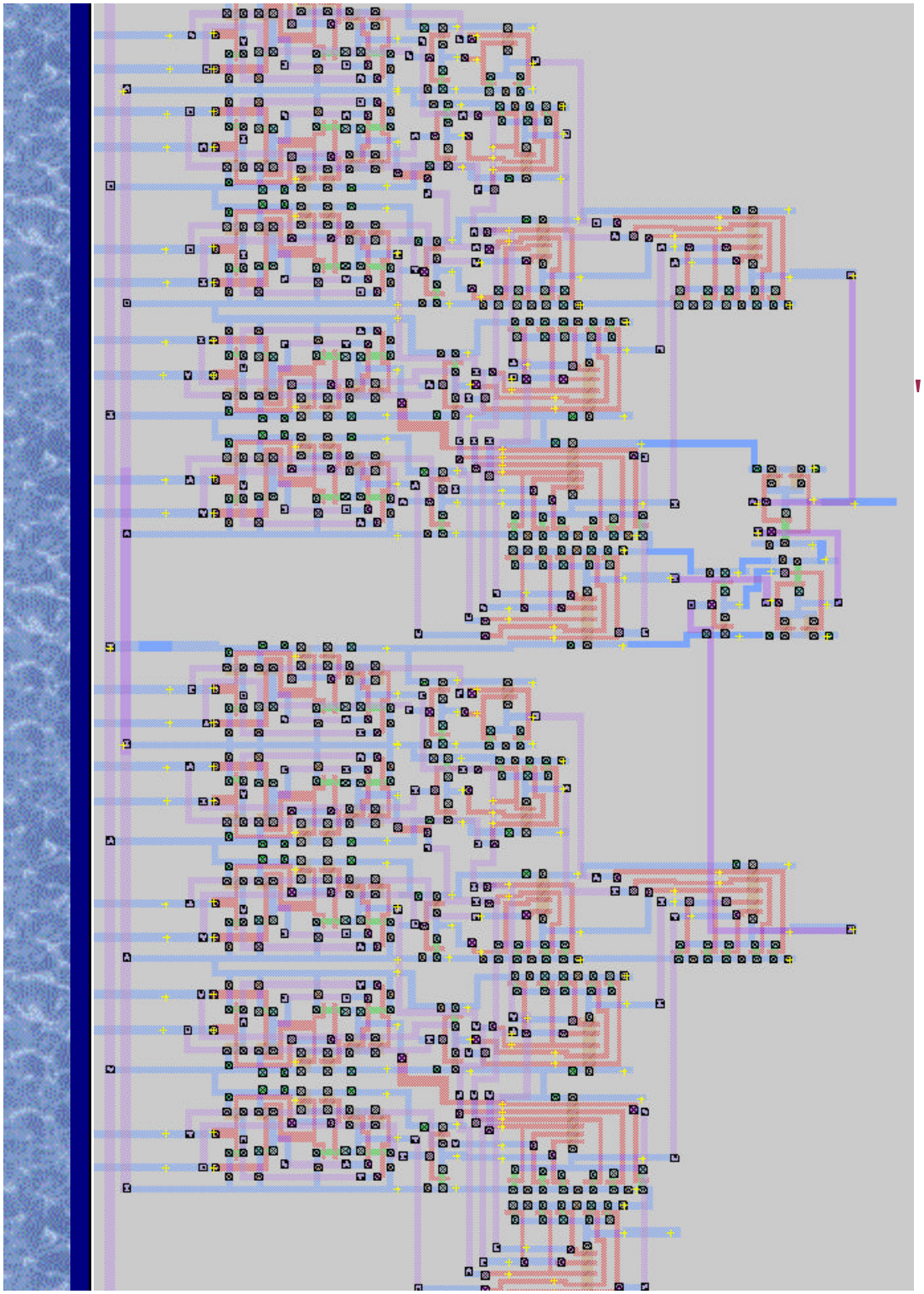
Register Cell

- Input runs horizontal at top
- Output runs horizontal at the bottom
- Power, ground, and control signals run vertically
- 12 transistors total



Register File

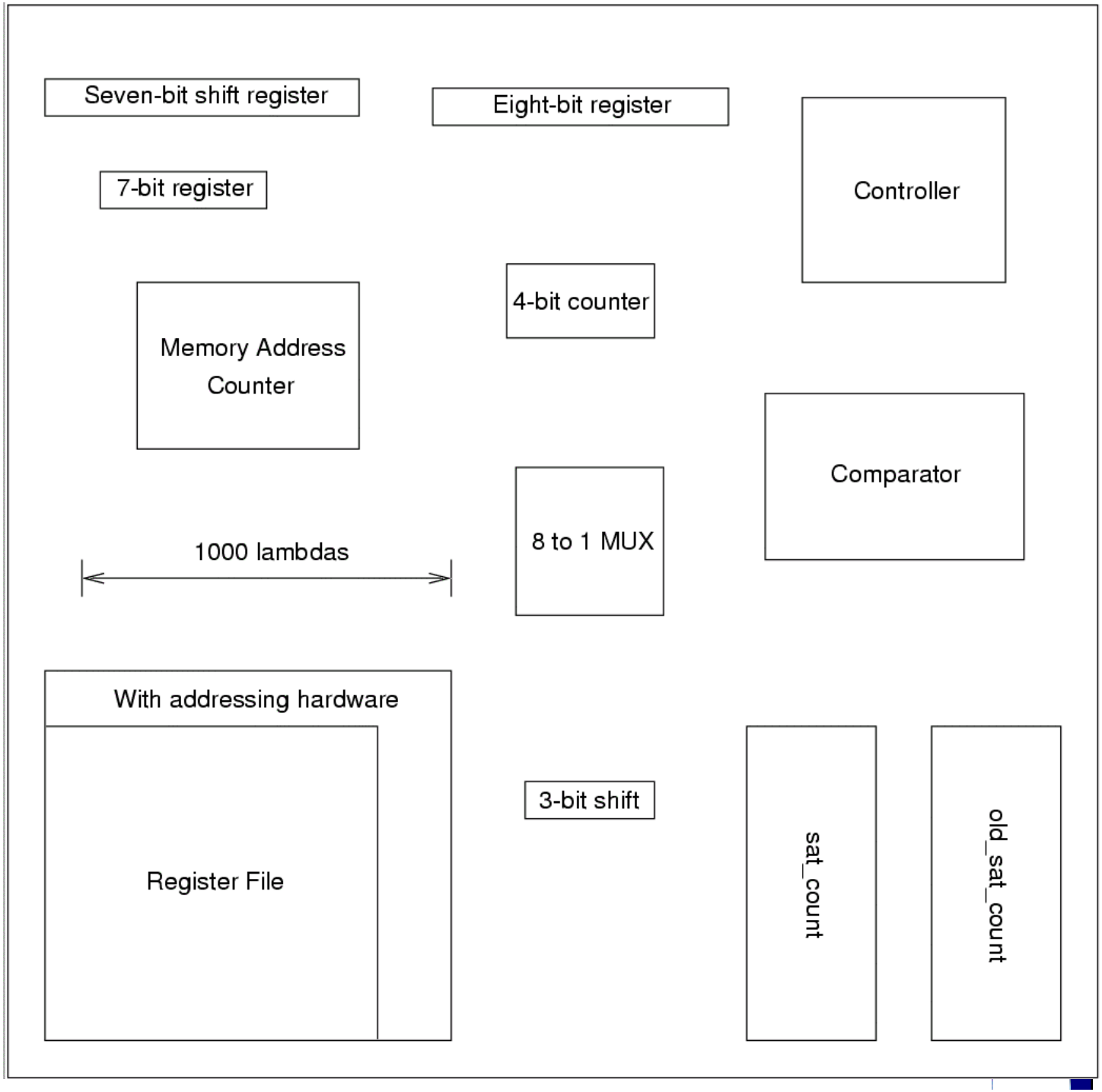




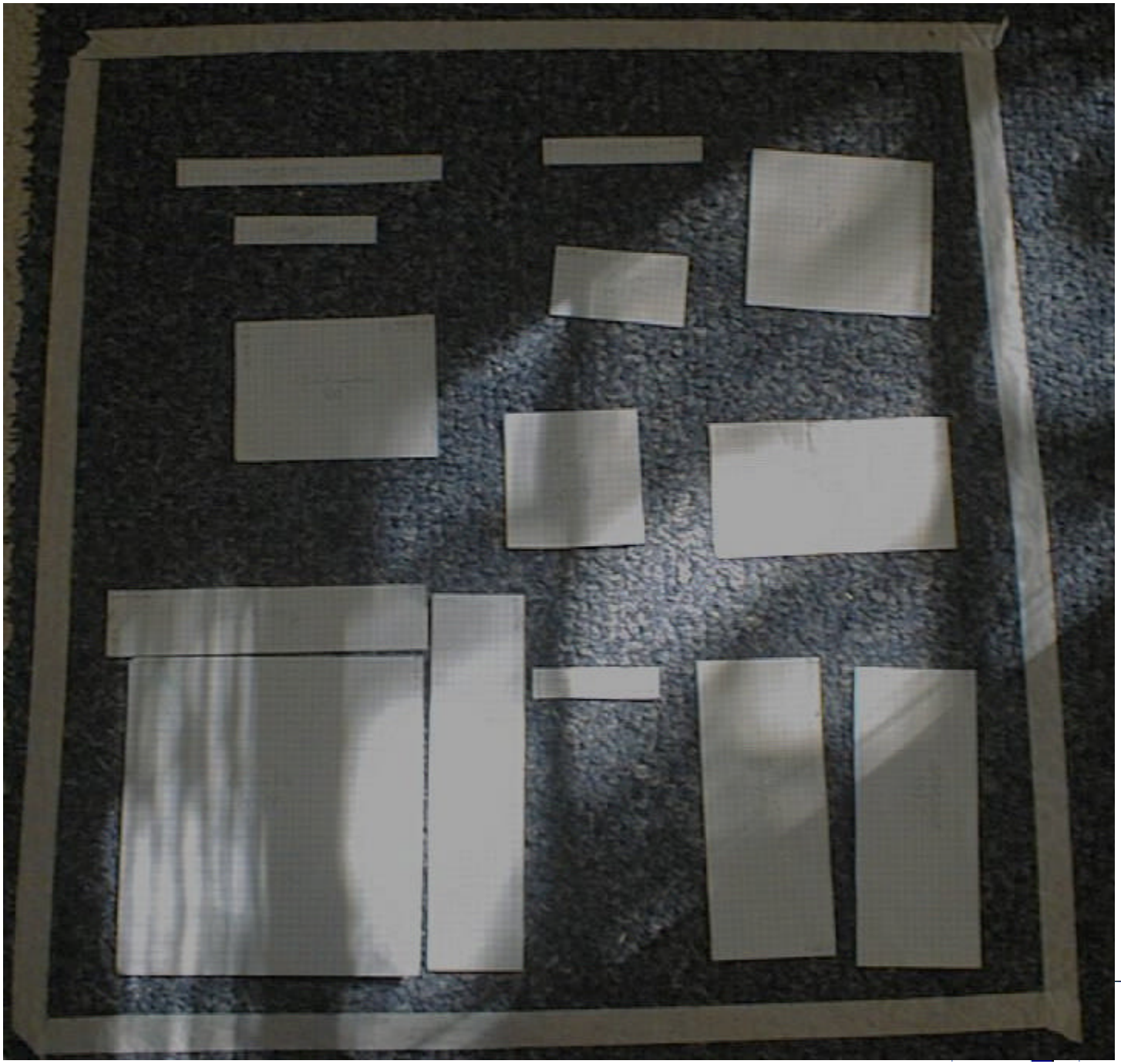
Ten-Bit Comparator

Used for comparing the number of satisfied clauses before and after a complement operation

Floorplan


Seven-bit shift register
Eight-bit register
Controller
7-bit register
4-bit counter
Memory Address Counter
Comparator
8 to 1 MUX
1000 lambdas
With addressing hardware
Register File
3-bit shift
sat_count
old_sat_count

The Plan on the Floor



Conclusion

- SAT is an important, computationally intensive problem.
- Our chip implements a heuristic algorithm to search for solutions.
- There is potential for multi-chip parallelism.