- Can your implementation solve the remaining 7 test  
cases (possibly with a longer time limit)?

|  |  |  |
| --- | --- | --- |
| adder.txt | O |  |
| arbiter.txt | O |  |
| bar.txt | O |  |
| cavlc.txt | O |  |
| ctrl.txt | O |  |
| dec.txt | O |  |
| i2c.txt | O |  |
| int2float.tx | O |  |
| max.txt | O |  |
| mem\_ctrl.txt | O |  |
| priority.txt | O |  |
| router.txt | O |  |
| sin.txt | O |  |
| random\_control/voter.aig | X | 1hr timeout |
| square.aig | O | 1hr timeout |
| sqrt.aig | X | 1hr timeout |
| multiplier.aig | X | 1hr timeout |
| log2.aig | X | 1hr timeout |
| hyp.aig | X | 1hr timeout |
| div.aig | X | 1hr timeout |

- How does your implementation compared to the BDD-  
based command `print\_unate` in ABC?

- What are the differences between random control and  
arithmetic circuits? Which category is more challenging?

Arithmetic circuits is more challenge