AIGER
AIGER

A language for describing hardware in AIGs

Literals are constants or signed variables rep. by positive ints
False is rep. by 0, True is rep. by 1
Variable is even number $\geq 2$, negated literal is odd number
Latches are assumed initialized to 0

Beginning of file is a header with format:

Header 1: aag M I L O A ⇐ text
Header 2: aig M I L O A ⇐ binary

M: maximum variable index
I: number of inputs
L: number of latches
O: Number of outputs
A: Number of and gates
All inputs are then listed, one per line, for example:

```
aag  7  2  0  2  3
2
4
6
...
```

Latches are then listed, one per line. Two literals: first is even, second is output. For example:

```
aag  7  2  0  2  3
...
2  3
...
```
The output literals are listed, one per line:

aag 3 2 0 1 1
...
6                   output 0
...

Then the and gates are listed, one per line. Three literals: even output, followed by two inputs
A single and gate:

aag 3 2 0 1 1
2                   input 1
4                   input 2
6                   output 3
6 2 4               and gate 0 1 \& 2

Finally an optional symbol table may be written followed by an optional comment section
Example: the empty circuit:
  aag 0 0 0 0 0

Example: the constant false:
  aag 0 0 0 1 0
  0

Example: the constant true:
  aag 0 0 0 1 0
  1

Example: a buffer:
  aag 1 1 0 1 0
  2          input
  2          output
Example: an inverter:

```
 aag 1 1 0 1 0

 2      input
 3      output

¬1
```

Example: an or gate:

```
 aag 3 2 0 1 1

2      input 1
4      input 2
7      output 3
6 3 5   and gate 1

¬(¬1 ∨ ¬2)

¬1 ∨ ¬2
```

Example: a toggle flip flop - one latch, two outputs:

```
 aag 1 0 1 2 0

2 3      latch showing next state literal
2        output 1
3        output 2
```
**Example: a half adder:**

<table>
<thead>
<tr>
<th>aag</th>
<th>7</th>
<th>2</th>
<th>0</th>
<th>2</th>
<th>3</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>12</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td></td>
<td>13</td>
<td>15</td>
<td></td>
<td></td>
</tr>
<tr>
<td>12</td>
<td></td>
<td>2</td>
<td>4</td>
<td></td>
<td></td>
</tr>
<tr>
<td>14</td>
<td></td>
<td>3</td>
<td>5</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- **header**
- **input 1**
- **input 2**
- **output 3**
- **output 6**
- **and gate 1**
- **and gate 2**
- **and gate 3**

- **input bit x**
- **input bit y**
- **sum bit**
- **carry**
- **x ⊕ y**
- **x ∧ y**
- **¬x ∧ ¬y**