// Do I need to constrain bits to [0,1] ?

$BIT == \{0, 1\}$
$EVENT ::= IN\langle\langle BIT\rangle\rangle \mid OUT\langle\langle BIT\rangle\rangle$
$Trace == seq\ EVENT$

SystemSkeleton

$tr : Trace$

We define a set of subtraces representing the output of a bit 3 times
and the input of that bit. For our entire trace there can only be an out
bit if there exists a subtrace where there the previous input is either an IN, or an IN OUT,
or IN OUT OUT. Here the exists s works to allow us to look at the last 1,2,3 entries of the trace.

ThreeOutBeforeIn

$tr : Trace$

$\forall t : Trace;\ b : BIT \mid tr = t \frown \langle OUT\ b\rangle \bullet$
$\quad \exists s : Trace \bullet$
$\quad\quad t = s \frown \langle IN\ b\rangle \lor$
$\quad\quad\quad t = s \frown \langle IN\ b, OUT\ b\rangle \lor$
$\quad\quad\quad t = s \frown \langle IN\ b, OUT\ b, OUT\ b\rangle$

OneInAfterThreeOut

$tr : Trace$

$\forall t : Trace;\ b : BIT \mid tr = t \frown \langle IN\ b\rangle \bullet$
$\quad t = \langle\rangle \lor (\exists s : Trace;\ c : BIT \bullet$
$\quad\quad t = s \frown \langle OUT\ c, OUT\ c, OUT\ c\rangle)$

The bit b can only be outputted if it was inputted 2 or 3 times prior(ly?)

ThreeInMajorityOut

$tr : Trace$

$\forall t : Trace;\ b : BIT \mid tr = t \frown \langle OUT\ b\rangle \bullet$
$\quad \exists s : Trace;\ c : BIT \bullet$
$\quad\quad t = s \frown \langle IN\ b, IN\ b, IN\ c\rangle \lor$
$\quad\quad\quad t = s \frown \langle IN\ b, IN\ c, IN\ b\rangle \lor$
$\quad\quad\quad t = s \frown \langle IN\ c, IN\ b, IN\ b\rangle$

ThreeInAfterOneOut

$tr : Trace$

$\forall t : Trace;\ b : BIT \mid tr = t \frown \langle IN\ b\rangle \bullet$
$\quad t = \langle\rangle \lor$
$\quad\quad \exists s : Trace;\ c, d, e : BIT \bullet$
$\quad\quad\quad t = s \frown \langle OUT\ c\rangle \lor$
$\quad\quad\quad t = s \frown \langle OUT\ c, IN\ d\rangle \lor$
$\quad\quad\quad t = s \frown \langle OUT\ c, IN\ d, IN\ e\rangle$

$ThreeOutOneIn == SystemSkeleton \land ThreeOutBeforeIn \land OneInAfterThreeOut$
$ThreeInOneMOut == SystemSkeleton \land ThreeInMajorityOut \land ThreeInAfterOneOut$

I had to peek for this. It's surprisingly elegant.
It states the count of all of the output bits (OUT 0, OUT1) should

at any given time be less than or equal to the input bits. (IN 0, IN 1)
Because ran OUT is the set {OUT 0, OUT 1}, there is not a restriction
stating that the output bit has to match the input bit.
This also allows for buffering.

Corruptor

$$tr : Trace$$

$$\#(tr \upharpoonright ran\ OUT) \leq \#(tr \upharpoonright ran\ IN)$$

In a system where no bits are corrupted, the number of INs matches the number of OUTs. To allow arbitrary corrupting, these two numbers must not match.