HWV axioms

HWV001-0.ax Connections, faults, and gates.
(connection(p_1, p_2) and value(p_1, v)) $\Rightarrow$ value(p_2, v)   cnf(value_propagation$_1$, axiom)
(connection(p_1, p_2) and value(p_2, v)) $\Rightarrow$ value(p_1, v)   cnf(value_propagation$_2$, axiom)
(value(p, v_1) and value(p, v_2)) $\Rightarrow$ equal_value(v_1, v_2)   cnf(unique_value, axiom)
$\neg$ equal_value(n_0, n_1)   cnf(equal_value$_1$, axiom)
$\neg$ equal_value(n_1, n_0)   cnf(equal_value$_2$, axiom)
mode(k, ok) $\Rightarrow$ $\neg$ mode(k, abnormal)   cnf(not_ok_and_abnormal, axiom)
type(k, any) $\Rightarrow$ (mode(k, ok) or mode(k, abnormal))   cnf(ok_or_abnormal, axiom)
(mode(k, ok) and type(k, and) and value(in(any, k), n_0)) $\Rightarrow$ value(out(n_1, k), n_0)   cnf(and_0x$_0$, axiom)
(mode(k, ok) and type(k, and) and value(in(n_1, k), n_1) and value(in(n_2, k), n_1)) $\Rightarrow$ value(out(n_1, k), n_1)   cnf(and_11$_1$, axio
(mode(k, ok) and type(k, and) and value(out(n_1, k), n_0)) $\Rightarrow$ (value(in(n_1, k), n_0) or value(in(n_2, k), n_0))   cnf(and_0$_{00}$, axio
(mode(k, ok) and type(k, and) and value(out(n_1, k), n_1)) $\Rightarrow$ value(in(n_1, k), n_1)   cnf(and_1_1x, axiom)
(mode(k, ok) and type(k, and) and value(out(n_1, k), n_1)) $\Rightarrow$ value(in(n_2, k), n_1)   cnf(and_1_x$_1$, axiom)
(mode(k, ok) and type(k, or) and value(in(any, k), n_1)) $\Rightarrow$ value(out(n_1, k), n_1)   cnf(or_1x$_1$, axiom)
(mode(k, ok) and type(k, or) and value(in(n_1, k), n_0) and value(in(n_2, k), n_0)) $\Rightarrow$ value(out(n_1, k), n_0)   cnf(or_00$_0$, axiom)
(mode(k, ok) and type(k, or) and value(out(n_1, k), n_1)) $\Rightarrow$ (value(in(n_1, k), n_1) or value(in(n_2, k), n_1))   cnf(or_1$_{11}$, axiom)
(mode(k, ok) and type(k, or) and value(out(n_1, k), n_0)) $\Rightarrow$ value(in(n_1, k), n_0)   cnf(or_0_0x, axiom)
(mode(k, ok) and type(k, or) and value(out(n_1, k), n_0)) $\Rightarrow$ value(in(n_2, k), n_0)   cnf(or_0$_{01}$, axiom)
(mode(k, ok) and type(k, not) and value(in(n_1, k), n_0)) $\Rightarrow$ value(out(n_1, k), n_1)   cnf(not_0_1_fw, axiom)
(mode(k, ok) and type(k, not) and value(in(n_1, k), n_1)) $\Rightarrow$ value(out(n_1, k), n_0)   cnf(not_1_0_fw, axiom)
(mode(k, ok) and type(k, not) and value(out(n_1, k), n_0)) $\Rightarrow$ value(in(n_1, k), n_1)   cnf(not_0_1_bw, axiom)
(mode(k, ok) and type(k, not) and value(out(n_1, k), n_1)) $\Rightarrow$ value(in(n_1, k), n_0)   cnf(not_1_0_bw, axiom)

HWV001-1.ax Half-adder.
type(x, halfadder) $\Rightarrow$ type(and$_1$(x), and)   cnf(halfadder_and$_1$, axiom)
type(x, halfadder) $\Rightarrow$ type(and$_2$(x), and)   cnf(halfadder_and$_2$, axiom)
type(x, halfadder) $\Rightarrow$ type(not$_1$(x), not)   cnf(halfadder_not$_1$, axiom)
type(x, halfadder) $\Rightarrow$ type(or$_1$(x), or)   cnf(halfadder_or$_1$, axiom)
type(x, halfadder) $\Rightarrow$ connection(in(n_1, x), in(n_1, or$_1$(x)))   cnf(halfadder_connection_in1_in1or$_1$, axiom)
type(x, halfadder) $\Rightarrow$ connection(in(n_2, x), in(n_2, or$_1$(x)))   cnf(halfadder_connection_in2_in2or$_1$, axiom)
type(x, halfadder) $\Rightarrow$ connection(in(n_1, x), in(n_1, and$_2$(x)))   cnf(halfadder_connection_in1_in1and$_2$, axiom)
type(x, halfadder) $\Rightarrow$ connection(in(n_2, x), in(n_2, and$_2$(x)))   cnf(halfadder_connection_in2_in2and$_2$, axiom)
type(x, halfadder) $\Rightarrow$ connection(out(s, x), out(n_1, and$_1$(x)))   cnf(halfadder_connection_outs_out1and$_1$, axiom)
type(x, halfadder) $\Rightarrow$ connection(out(c, x), out(n_1, and$_2$(x)))   cnf(halfadder_connection_outc_out1and$_2$, axiom)
type(x, halfadder) $\Rightarrow$ connection(out(n_1, or$_1$(x)), in(n_1, and$_1$(x)))   cnf(halfadder_connection_out1or1_in1_and$_1$, axiom)
type(x, halfadder) $\Rightarrow$ connection(out(n_1, and$_2$(x)), in(n_1, not$_1$(x)))   cnf(halfadder_connection_out1and2_in1not$_1$, axiom)
type(x, halfadder) $\Rightarrow$ connection(out(n_1, not$_1$(x)), in(n_2, and$_1$(x)))   cnf(halfadder_connection_out1not1_in2and$_1$, axiom)

HWV001-2.ax Full-adder.
type(x, fulladder) $\Rightarrow$ type(h_1(x), halfadder)   cnf(fulladder_halfadder$_1$, axiom)
type(x, fulladder) $\Rightarrow$ type(h_2(x), halfadder)   cnf(fulladder_halfadder$_2$, axiom)
type(x, fulladder) $\Rightarrow$ type(or$_1$(x), or)   cnf(fulladder_or$_1$, axiom)
type(x, fulladder) $\Rightarrow$ connection(out($s, h_1(x)$), in($n_2, h_2(x)$))   cnf(fulladder_connection_outsh1_in2h$_2$, axiom)
type(x, fulladder) $\Rightarrow$ connection(out($c, h_1(x)$), in(n_2, or$_1$(x)))   cnf(fulladder_connection_outch1_in2or$_1$, axiom)
type(x, fulladder) $\Rightarrow$ connection(out($c, h_2(x)$), in(n_1, or$_1$(x)))   cnf(fulladder_connection_outch2_in1or$_1$, axiom)
type(x, fulladder) $\Rightarrow$ connection(in(n_1, x), in($n_1, h_2(x)$))   cnf(fulladder_connection_in1_in1h$_2$, axiom)
type(x, fulladder) $\Rightarrow$ connection(in(n_2, x), in($n_1, h_1(x)$))   cnf(fulladder_connection_in2_in1h$_1$, axiom)
type(x, fulladder) $\Rightarrow$ connection(in(c, x), in($n_2, h_1(x)$))   cnf(fulladder_connection_inc_in2h$_1$, axiom)
type(x, fulladder) $\Rightarrow$ connection(out(s, x), out($s, h_2(x)$))   cnf(fulladder_connection_outs_outsh$_2$, axiom)
type(x, fulladder) $\Rightarrow$ connection(out(c, x), out(n_1, or$_1$(x)))   cnf(fulladder_connection_outc_out1or$_1$, axiom)

HWV002-0.ax Connections, faults, and gates.
(connection(p_1, p_2) and 0) $\Rightarrow$ 0   cnf(value_propagation_zero$_1$, axiom)
(connection(p_1, p_2) and 1) $\Rightarrow$ 1   cnf(value_propagation_one$_1$, axiom)
(connection(p_1, p_2) and 0) $\Rightarrow$ 0   cnf(value_propagation_zero$_2$, axiom)
(connection(p_1, p_2) and 1) $\Rightarrow$ 1   cnf(value_propagation_one$_2$, axiom)
0 $\Rightarrow$ $\neg$ 1   cnf(unique_value, axiom)
(and_ok(k) and 0) $\Rightarrow$ 0   cnf(and_0x$_0$, axiom)

(and_ok(k) and 0) $\Rightarrow$ 0 cnf(and_x0$_0$, axiom)
(and_ok(k) and 1 and 1) $\Rightarrow$ 1 cnf(and_11$_1$, axiom)
(and_ok(k) and 0) $\Rightarrow$ (0 or 0) cnf(and_0$_{00}$, axiom)
(and_ok(k) and 1) $\Rightarrow$ 1 cnf(and_1_1x, axiom)
(and_ok(k) and 1) $\Rightarrow$ 1 cnf(and_1_x$_1$, axiom)
and_ok(k) $\Rightarrow$ $\neg$ abnormal(k) cnf(not_and_ok_and_abnormal, axiom)
logic_and(k) $\Rightarrow$ (and_ok(k) or abnormal(k)) cnf(and_ok_or_abnormal, axiom)
(or_ok(k) and 1) $\Rightarrow$ 1 cnf(or_1x$_1$, axiom)
(or_ok(k) and 1) $\Rightarrow$ 1 cnf(or_x1$_1$, axiom)
(or_ok(k) and 0 and 0) $\Rightarrow$ 0 cnf(or_00$_0$, axiom)
(or_ok(k) and 1) $\Rightarrow$ (1 or 1) cnf(or_1$_{11}$, axiom)
(or_ok(k) and 0) $\Rightarrow$ 0 cnf(or_0_0x, axiom)
(or_ok(k) and 0) $\Rightarrow$ 0 cnf(or_0$_{01}$, axiom)
or_ok(k) $\Rightarrow$ $\neg$ abnormal(k) cnf(not_or_ok_and_abnormal, axiom)
logic_or(k) $\Rightarrow$ (or_ok(k) or abnormal(k)) cnf(or_ok_or_abnormal, axiom)
(not_ok(k) and 0) $\Rightarrow$ 1 cnf(not_0_1_fw, axiom)
(not_ok(k) and 1) $\Rightarrow$ 0 cnf(not_1_0_fw, axiom)
(not_ok(k) and 0) $\Rightarrow$ 1 cnf(not_0_1_bw, axiom)
(not_ok(k) and 1) $\Rightarrow$ 0 cnf(not_1_0_bw, axiom)
not_ok(k) $\Rightarrow$ $\neg$ abnormal(k) cnf(not__not_ok_and_abnormal, axiom)
logic_not(k) $\Rightarrow$ (not_ok(k) or abnormal(k)) cnf(not_ok_or_abnormal, axiom)

HWV002-1.ax Half-adder.
halfadder(x) $\Rightarrow$ logic_and(and$_1$(x)) cnf(halfadder_and$_1$, axiom)
halfadder(x) $\Rightarrow$ logic_and(and$_2$(x)) cnf(halfadder_and$_2$, axiom)
halfadder(x) $\Rightarrow$ logic_not(not$_1$(x)) cnf(halfadder_not$_1$, axiom)
halfadder(x) $\Rightarrow$ logic_or(or$_1$(x)) cnf(halfadder_or$_1$, axiom)
halfadder(x) $\Rightarrow$ connection(in$_1$(x), in$_1$(or$_1$(x))) cnf(halfadder_connection_in1_in1or$_1$, axiom)
halfadder(x) $\Rightarrow$ connection(in$_2$(x), in$_2$(or$_1$(x))) cnf(halfadder_connection_in2_in2or$_1$, axiom)
halfadder(x) $\Rightarrow$ connection(in$_1$(x), in$_1$(and$_2$(x))) cnf(halfadder_connection_in1_in1and$_2$, axiom)
halfadder(x) $\Rightarrow$ connection(in$_2$(x), in$_2$(and$_2$(x))) cnf(halfadder_connection_in2_in2and$_2$, axiom)
halfadder(x) $\Rightarrow$ connection(outs(x), out$_1$(and$_1$(x))) cnf(halfadder_connection_outs_out1and$_1$, axiom)
halfadder(x) $\Rightarrow$ connection(outc(x), out$_1$(and$_2$(x))) cnf(halfadder_connection_outc_out1and$_2$, axiom)
halfadder(x) $\Rightarrow$ connection(out$_1$(or$_1$(x)), in$_1$(and$_1$(x))) cnf(halfadder_connection_out1or1_in1_and$_1$, axiom)
halfadder(x) $\Rightarrow$ connection(out$_1$(and$_2$(x)), in$_1$(not$_1$(x))) cnf(halfadder_connection_out1and2_in1not$_1$, axiom)
halfadder(x) $\Rightarrow$ connection(out$_1$(not$_1$(x)), in$_2$(and$_1$(x))) cnf(halfadder_connection_out1not1_in2and$_1$, axiom)

HWV002-2.ax Full-adder.
fulladder(x) $\Rightarrow$ halfadder(h_1(x)) cnf(fulladder_halfadder$_1$, axiom)
fulladder(x) $\Rightarrow$ halfadder(h_2(x)) cnf(fulladder_halfadder$_2$, axiom)
fulladder(x) $\Rightarrow$ logic_or(or$_1$(x)) cnf(fulladder_or$_1$, axiom)
fulladder(x) $\Rightarrow$ connection(outs(h_1(x)), in$_2$(h_2(x))) cnf(fulladder_connection_outsh1_in2h$_2$, axiom)
fulladder(x) $\Rightarrow$ connection(outc(h_1(x)), in$_2$(or$_1$(x))) cnf(fulladder_connection_outch1_in2or$_1$, axiom)
fulladder(x) $\Rightarrow$ connection(outc(h_2(x)), in$_1$(or$_1$(x))) cnf(fulladder_connection_outch2_in1or$_1$, axiom)
fulladder(x) $\Rightarrow$ connection(in$_1$(x), in$_1$(h_2(x))) cnf(fulladder_connection_in1_in1h$_2$, axiom)
fulladder(x) $\Rightarrow$ connection(in$_2$(x), in$_1$(h_1(x))) cnf(fulladder_connection_in2_in1h$_1$, axiom)
fulladder(x) $\Rightarrow$ connection(inc(x), in$_2$(h_1(x))) cnf(fulladder_connection_inc_in2h$_1$, axiom)
fulladder(x) $\Rightarrow$ connection(outs(x), outs(h_2(x))) cnf(fulladder_connection_outs_outsh$_2$, axiom)
fulladder(x) $\Rightarrow$ connection(outc(x), out$_1$(or$_1$(x))) cnf(fulladder_connection_outc_out1or$_1$, axiom)

HWV problems

HWV001-1.p Interchange inputs to outputs
This validates one of the circuit designs that will take x and y as input and output y and x without crossing any wires.
include('Axioms/HWC002-0.ax')
and(x, y) = and(y, x) cnf(and_commutativity, axiom)
or(x, y) = or(y, x) cnf(or_commutativity, axiom)
and(x, and(y, z)) = and(and(x, y), z) cnf(and_associativity, axiom)
or(x, or(y, z)) = or(or(x, y), z) cnf(or_associativity, axiom)

$\mathrm{or}(\mathrm{or}(x,y),z) = \mathrm{or}(\mathrm{or}(x,z),y)$ cnf(or_sorting, axiom)
$\mathrm{and}(\mathrm{and}(x,y),z) = \mathrm{and}(\mathrm{and}(x,z),y)$ cnf(and_sorting, axiom)
$\mathrm{not}(\mathrm{and}(x,y)) = \mathrm{or}(\mathrm{not}(x),\mathrm{not}(y))$ cnf(not_canonicalization$_1$, axiom)
$\mathrm{not}(\mathrm{or}(x,y)) = \mathrm{and}(\mathrm{not}(x),\mathrm{not}(y))$ cnf(not_canonicalization$_2$, axiom)
$\mathrm{and}(\mathrm{or}(x,y),z) = \mathrm{or}(\mathrm{and}(x,z),\mathrm{and}(y,z))$ cnf(and_or_canonicalization, axiom)
$\mathrm{and}(x,x) = x$ cnf(and_simplification$_1$, axiom)
$\mathrm{and}(\mathrm{and}(x,y),y) = \mathrm{and}(x,y)$ cnf(and_simplification$_2$, axiom)
$\mathrm{and}(\mathrm{and}(x,y),x) = \mathrm{and}(x,y)$ cnf(and_simplification$_3$, axiom)
$\mathrm{or}(x,x) = x$ cnf(or_simplification$_1$, axiom)
$\mathrm{or}(\mathrm{or}(x,y),y) = \mathrm{or}(x,y)$ cnf(or_simplification$_2$, axiom)
$\mathrm{or}(\mathrm{or}(x,y),x) = \mathrm{or}(x,y)$ cnf(or_simplification$_3$, axiom)
$\mathrm{and}(x,\mathrm{not}(x)) = n_0$ cnf(and_not_simplification$_1$, axiom)
$\mathrm{and}(\mathrm{and}(x,y),\mathrm{not}(y)) = n_0$ cnf(and_not_simplification$_2$, axiom)
$\mathrm{and}(\mathrm{and}(x,y),\mathrm{not}(x)) = n_0$ cnf(and_not_simplification$_3$, axiom)
$\mathrm{or}(x,\mathrm{not}(x)) = n_1$ cnf(or_not_simplification$_1$, axiom)
$\mathrm{or}(\mathrm{or}(x,y),\mathrm{not}(y)) = n_1$ cnf(or_not_simplification$_2$, axiom)
$\mathrm{or}(\mathrm{or}(x,y),\mathrm{not}(x)) = n_1$ cnf(or_not_simplification$_3$, axiom)
$\mathrm{not}(\mathrm{not}(x)) = x$ cnf(not_simplification, axiom)
$\mathrm{or}(\mathrm{and}(x,y),\mathrm{and}(x,\mathrm{not}(y))) = x$ cnf(and_or_not_simplification$_1$, axiom)
$\mathrm{or}(\mathrm{and}(x,y),\mathrm{and}(y,\mathrm{not}(x))) = y$ cnf(and_or_not_simplification$_2$, axiom)
$a_1 = \mathrm{and}(b_1,b_3)$ cnf(constructor$_1$, negated_conjecture)
$a_2 = \mathrm{and}(b_2,b_3)$ cnf(constructor$_2$, negated_conjecture)
$b_1 = \mathrm{not}(d_1)$ cnf(constructor$_3$, negated_conjecture)
$b_2 = \mathrm{not}(d_2)$ cnf(constructor$_4$, negated_conjecture)
$b_3 = \mathrm{or}(c_1,c_2)$ cnf(constructor$_5$, negated_conjecture)
$c_1 = \mathrm{or}(d_1,d_3)$ cnf(constructor$_6$, negated_conjecture)
$c_2 = \mathrm{or}(d_2,d_3)$ cnf(constructor$_7$, negated_conjecture)
$d_3 = f_3$ cnf(constructor$_8$, negated_conjecture)
$d_1 = \mathrm{not}(e_1)$ cnf(constructor$_9$, negated_conjecture)
$d_2 = \mathrm{not}(e_2)$ cnf(constructor$_{10}$, negated_conjecture)
$e_1 = \mathrm{or}(f_1,f_3)$ cnf(constructor$_{11}$, negated_conjecture)
$e_2 = \mathrm{or}(f_2,f_3)$ cnf(constructor$_{12}$, negated_conjecture)
$f_1 = \mathrm{not}(i_1)$ cnf(constructor$_{13}$, negated_conjecture)
$f_2 = \mathrm{not}(i_2)$ cnf(constructor$_{14}$, negated_conjecture)
$f_3 = \mathrm{and}(i_1,i_2)$ cnf(constructor$_{15}$, negated_conjecture)
$\mathrm{circuit}(\mathrm{input}(i_1,i_2),\mathrm{output}(a_1,a_2))$ cnf(circuit_description, negated_conjecture)
$\neg\,\mathrm{circuit}(\mathrm{input}(i_1,i_2),\mathrm{output}(i_2,i_1))$ cnf(prove_interchange, negated_conjecture)

HWV003-1.p One bit Full Adder
include('Axioms/HWC002-0.ax')
$\mathrm{not}(\mathrm{and}(x,y)) = \mathrm{or}(\mathrm{not}(x),\mathrm{not}(y))$ cnf(demorgan$_1$, axiom)
$\mathrm{not}(\mathrm{or}(x,y)) = \mathrm{and}(\mathrm{not}(x),\mathrm{not}(y))$ cnf(demorgan$_2$, axiom)
$\mathrm{not}(\mathrm{not}(x)) = x$ cnf(not_involution, axiom)
$\mathrm{and}(x,y) = \mathrm{and}(y,x)$ cnf(and_symmetry, negated_conjecture)
$\mathrm{or}(x,y) = \mathrm{or}(y,x)$ cnf(or_symmetry, negated_conjecture)
$\mathrm{and}(\mathrm{or}(x,y),z) = \mathrm{or}(\mathrm{and}(x,z),\mathrm{and}(y,z))$ cnf(and_or_simplification, negated_conjecture)
$\mathrm{and}(\mathrm{and}(x,y),z) = \mathrm{and}(\mathrm{and}(x,z),y)$ cnf(and_commutativity, negated_conjecture)
$\mathrm{or}(\mathrm{or}(x,y),z) = \mathrm{or}(\mathrm{or}(x,z),y)$ cnf(or_commutativity, negated_conjecture)
$\mathrm{and}(x,\mathrm{not}(x)) = n_0$ cnf(and_not_evaluation$_1$, axiom)
$\mathrm{or}(x,\mathrm{not}(x)) = n_1$ cnf(or_not_evaluation$_1$, axiom)
$\mathrm{and}(x,x) = x$ cnf(and_idempotency, axiom)
$\mathrm{or}(x,x) = x$ cnf(or_idempotency, axiom)
$\mathrm{and}(\mathrm{and}(x,y),\mathrm{not}(y)) = n_0$ cnf(and_not_evaluation$_2$, axiom)
$\mathrm{and}(\mathrm{and}(x,y),\mathrm{not}(x)) = n_0$ cnf(and_not_evaluation$_3$, axiom)
$\mathrm{or}(\mathrm{or}(x,y),\mathrm{not}(y)) = n_1$ cnf(or_not_evaluation$_2$, axiom)
$\mathrm{or}(\mathrm{or}(x,y),\mathrm{not}(x)) = n_1$ cnf(or_not_evaluation$_3$, axiom)
$\mathrm{and}(\mathrm{and}(x,y),y) = \mathrm{and}(x,y)$ cnf(and_evaluation$_1$, axiom)
$\mathrm{or}(\mathrm{or}(x,y),y) = \mathrm{or}(x,y)$ cnf(or_evaluation$_1$, axiom)
$\mathrm{and}(\mathrm{and}(\mathrm{and}(x_1,x_2),x_3),\mathrm{not}(x_1)) = n_0$ cnf(and_not_evaluation$_4$, axiom)
$\mathrm{and}(\mathrm{and}(\mathrm{and}(x_1,x_2),x_3),\mathrm{not}(x_2)) = n_0$ cnf(and_not_evaluation$_5$, axiom)

$\mathrm{or}(\mathrm{and}(x, y), y) = y$ cnf(and_or_subsumption$_1$, axiom)
$\mathrm{or}(\mathrm{and}(x, y), x) = x$ cnf(and_or_subsumption$_2$, axiom)
$\mathrm{or}(\mathrm{or}(\mathrm{and}(x, y), z), y) = \mathrm{or}(z, y)$ cnf(and_or_subsumption$_3$, axiom)
$\mathrm{or}(\mathrm{or}(x, \mathrm{and}(y, z)), z) = \mathrm{or}(x, z)$ cnf(and_or_subsumption$_4$, axiom)
$\mathrm{or}(\mathrm{and}(x, \mathrm{not}(y)), y) = \mathrm{or}(x, y)$ cnf(karnaugh$_1$, axiom)
$\mathrm{or}(\mathrm{and}(\mathrm{not}(x), \mathrm{not}(y)), y) = \mathrm{or}(y, \mathrm{not}(x))$ cnf(karnaugh$_2$, axiom)
$\mathrm{or}(\mathrm{and}(\mathrm{and}(x, y), \mathrm{not}(z)), \mathrm{and}(x, z)) = \mathrm{or}(\mathrm{and}(x, y), \mathrm{and}(x, z))$ cnf(karnaugh$_3$, axiom)
$\mathrm{xor}(x, y) = \mathrm{or}(\mathrm{and}(x, \mathrm{not}(y)), \mathrm{and}(y, \mathrm{not}(x)))$ cnf(xor_definition, axiom)
$\mathrm{carryout}(x, y, z) = \mathrm{or}(\mathrm{and}(x, \mathrm{or}(y, z)), \mathrm{and}(\mathrm{not}(x), \mathrm{and}(y, z)))$ cnf(carryout_definition, negated_conjecture)
$x + y{=}z = \mathrm{xor}(\mathrm{xor}(x, y), z)$ cnf(sum_definition, negated_conjecture)
$a_{11} = \mathrm{not}(\mathrm{and}(a, b))$ cnf(circuit$_1$, negated_conjecture)
$a_{12} = \mathrm{not}(\mathrm{and}(a_{11}, a))$ cnf(circuit$_2$, negated_conjecture)
$a_{13} = \mathrm{not}(\mathrm{and}(a_{11}, b))$ cnf(circuit$_3$, negated_conjecture)
$a_{14} = \mathrm{not}(\mathrm{and}(a_{12}, a_{13}))$ cnf(circuit$_4$, negated_conjecture)
$a_{15} = \mathrm{not}(\mathrm{and}(a_{14}, \mathrm{carryin}))$ cnf(circuit$_5$, negated_conjecture)
$a_{16} = \mathrm{not}(\mathrm{and}(a_{14}, a_{15}))$ cnf(circuit$_6$, negated_conjecture)
$a_{17} = \mathrm{not}(\mathrm{and}(a_{15}, \mathrm{carryin}))$ cnf(circuit$_7$, negated_conjecture)
$s_1 = \mathrm{not}(\mathrm{and}(a_{16}, a_{17}))$ cnf(circuit$_8$, negated_conjecture)
$c_1 = \mathrm{not}(\mathrm{and}(a_{11}, a_{15}))$ cnf(circuit$_9$, negated_conjecture)
$\mathrm{circuit}(s_1, c_1)$ cnf(the_output_circuit, negated_conjecture)
$\neg\, \mathrm{circuit}(a + b{=}\mathrm{carryin}, \mathrm{carryout}(a, b, \mathrm{carryin}))$ cnf(prove_circuit, negated_conjecture)

HWV003-2.p One bit Full Adder
Prove that the given implementation of a one-bit full adder using only NAND gates is correct
$\mathrm{or}(x, y) = \mathrm{or}(y, x)$ cnf(or_commutative, axiom)
$\mathrm{and}(x, y) = \mathrm{and}(y, x)$ cnf(and_commutative, axiom)
$\mathrm{or}(\mathrm{or}(x, y), z) = \mathrm{or}(x, \mathrm{or}(y, z))$ cnf(or_associative, axiom)
$\mathrm{and}(\mathrm{and}(x, y), z) = \mathrm{and}(x, \mathrm{and}(y, z))$ cnf(and_associative, axiom)
$\mathrm{or}(x, \mathrm{ll}_0) = x$ cnf(or_identity, axiom)
$\mathrm{and}(x, \mathrm{ll}_1) = x$ cnf(and_identity, axiom)
$\mathrm{and}(x, \mathrm{or}(y, z)) = \mathrm{or}(\mathrm{and}(x, y), \mathrm{and}(x, z))$ cnf(or_distributive, axiom)
$\mathrm{or}(x, \mathrm{and}(y, z)) = \mathrm{and}(\mathrm{or}(x, y), \mathrm{or}(x, z))$ cnf(and_distributive, axiom)
$\mathrm{or}(x, \mathrm{not}(x)) = \mathrm{ll}_1$ cnf(or_complement, axiom)
$\mathrm{and}(x, \mathrm{not}(x)) = \mathrm{ll}_0$ cnf(and_complement, axiom)
$\mathrm{or}(x, x) = x$ cnf(or_idempotent, axiom)
$\mathrm{and}(x, x) = x$ cnf(and_idempotent, axiom)
$\mathrm{not}(\mathrm{not}(x)) = x$ cnf(not_involution, axiom)
$\mathrm{not}(\mathrm{ll}_0) = \mathrm{ll}_1$ cnf(ll0_inverse, axiom)
$\mathrm{not}(\mathrm{ll}_1) = \mathrm{ll}_0$ cnf(ll1_inverse, axiom)
$\mathrm{or}(x, \mathrm{ll}_1) = \mathrm{ll}_1$ cnf(or_boundedness, axiom)
$\mathrm{and}(x, \mathrm{ll}_0) = \mathrm{ll}_0$ cnf(and_boundedness, axiom)
$\mathrm{or}(x, \mathrm{and}(x, y)) = x$ cnf(or_absorption, axiom)
$\mathrm{and}(x, \mathrm{or}(x, y)) = x$ cnf(and_absorption, axiom)
$\mathrm{not}(\mathrm{or}(x, y)) = \mathrm{and}(\mathrm{not}(x), \mathrm{not}(y))$ cnf(or_demorgan, axiom)
$\mathrm{not}(\mathrm{and}(x, y)) = \mathrm{or}(\mathrm{not}(x), \mathrm{not}(y))$ cnf(and_demorgan, axiom)
$\mathrm{or}(\mathrm{and}(\mathrm{not}(x), y), \mathrm{and}(x, \mathrm{not}(y))) = \mathrm{xor}(x, y)$ cnf(xor_definition, axiom)
$\mathrm{xor}(x, y) = \mathrm{xor}(y, x)$ cnf(xor_commutative, axiom)
$\mathrm{xor}(x, \mathrm{xor}(y, z)) = \mathrm{xor}(\mathrm{xor}(x, y), z)$ cnf(xor_associative, axiom)
$\mathrm{xor}(\mathrm{xor}(a, b), \mathrm{cin}) = \mathrm{sum\_def}$ cnf(sum_def, negated_conjecture)
$\mathrm{or}(\mathrm{and}(\mathrm{cin}, \mathrm{or}(a, b)), \mathrm{and}(\mathrm{not}(\mathrm{cin}), \mathrm{and}(a, b))) = \mathrm{carry\_def}$ cnf(carry_def, negated_conjecture)
$\mathrm{not}(\mathrm{and}(a, b)) = t_1$ cnf(t_1, negated_conjecture)
$\mathrm{not}(\mathrm{and}(a, t_1)) = t_2$ cnf(t_2, negated_conjecture)
$\mathrm{not}(\mathrm{and}(b, t_1)) = t_3$ cnf(t_3, negated_conjecture)
$\mathrm{not}(\mathrm{and}(t_2, t_3)) = t_4$ cnf(t_4, negated_conjecture)
$\mathrm{not}(\mathrm{and}(t_4, \mathrm{cin})) = t_5$ cnf(t_5, negated_conjecture)
$\mathrm{not}(\mathrm{and}(t_4, t_5)) = t_6$ cnf(t_6, negated_conjecture)
$\mathrm{not}(\mathrm{and}(t_5, \mathrm{cin})) = t_7$ cnf(t_7, negated_conjecture)
$\mathrm{not}(\mathrm{and}(t_6, t_7)) = +$ cnf(sum, negated_conjecture)
$\mathrm{not}(\mathrm{and}(t_1, t_5)) = \mathrm{carry}$ cnf(carry, negated_conjecture)
$+ = \mathrm{sum\_def} \;\Rightarrow\; \mathrm{carry} \neq \mathrm{carry\_def}$ cnf(prove_circuit, negated_conjecture)

HWV004-1.p Two bit Full Adder
include('Axioms/HWC002-0.ax')
not(and(x, y)) = or(not(x), not(y))   cnf(demorgan$_1$, axiom)
not(or(x, y)) = and(not(x), not(y))   cnf(demorgan$_2$, axiom)
not(not(x)) = x cnf(not_involution, axiom)
and(x, y) = and(y, x)   cnf(and_symmetry, negated_conjecture)
or(x, y) = or(y, x)   cnf(or_symmetry, negated_conjecture)
and(or(x, y), z) = or(and(x, z), and(y, z))   cnf(and_or_simplification, negated_conjecture)
and(and(x, y), z) = and(and(x, z), y)   cnf(and_commutativity, negated_conjecture)
or(or(x, y), z) = or(or(x, z), y)   cnf(or_commutativity, negated_conjecture)
and(x, not(x)) = n_0 cnf(and_not_evaluation$_1$, axiom)
or(x, not(x)) = n_1 cnf(or_not_evaluation$_1$, axiom)
and(x, x) = x cnf(and_idempotency, axiom)
or(x, x) = x cnf(or_idempotency, axiom)
and(and(x, y), not(y)) = n_0 cnf(and_not_evaluation$_2$, axiom)
and(and(x, y), not(x)) = n_0 cnf(and_not_evaluation$_3$, axiom)
or(or(x, y), not(y)) = n_1 cnf(or_not_evaluation$_2$, axiom)
or(or(x, y), not(x)) = n_1 cnf(or_not_evaluation$_3$, axiom)
and(and(x, y), y) = and(x, y)   cnf(and_evaluation$_1$, axiom)
or(or(x, y), y) = or(x, y)   cnf(or_evaluation$_1$, axiom)
and(and(and(x_1, x_2), x_3), not(x_1)) = n_0 cnf(and_not_evaluation$_4$, axiom)
and(and(and(x_1, x_2), x_3), not(x_2)) = n_0 cnf(and_not_evaluation$_5$, axiom)
or(and(x, y), y) = y cnf(and_or_subsumption$_1$, axiom)
or(and(x, y), x) = x cnf(and_or_subsumption$_2$, axiom)
or(or(and(x, y), z), y) = or(z, y)   cnf(and_or_subsumption$_3$, axiom)
or(or(x, and(y, z)), z) = or(x, z)   cnf(and_or_subsumption$_4$, axiom)
or(and(x, not(y)), y) = or(x, y)   cnf(karnaugh$_1$, axiom)
or(and(not(x), not(y)), y) = or(y, not(x))   cnf(karnaugh$_2$, axiom)
or(and(and(x, y), not(z)), and(x, z)) = or(and(x, y), and(x, z))   cnf(karnaugh$_3$, axiom)
xor(x, y) = or(and(x, not(y)), and(y, not(x)))   cnf(xor_definition, axiom)
carryout(x, y, z) = or(and(x, or(y, z)), and(not(x), and(y, z)))   cnf(carryout_definition, negated_conjecture)
$x + y$=z = xor(xor(x, y), z)   cnf(sum_definition, negated_conjecture)
$s_0 = a_0 + b_0$=n_0 cnf(s0_definition, negated_conjecture)
$s_1 = a_1 + b_1$=carryout(a_0, b_0, n_0)   cnf(s1_definition, negated_conjecture)
overflow = carryout(a_1, b_1, carryout(a_0, b_0, n_0))   cnf(overflow_definition, negated_conjecture)
circuit(s_0, s_1, overflow)   cnf(the_output_circuit, negated_conjecture)
$\neg$ circuit(xor(a_0, b_0), xor(xor(a_1, b_1), carryout(a_0, b_0, n_0)), or(and(a_1, b_1), and(and(a_0, b_0), or(a_1, b_1))))   cnf(prove_circuit, neg

HWV005-1.p A halfadder built from and, or and not gates
include('Axioms/HWV001-0.ax')
include('Axioms/HWV001-1.ax')
type(h, halfadder)   cnf(h_isa_halfadder, hypothesis)
value(in(n_1, h), n_1)   cnf(in1$_1$, hypothesis)
value(in(n_2, h), n_0)   cnf(in2$_0$, hypothesis)
value(out(s, h), n_0)   cnf(outs$_0$, hypothesis)
value(out(c, h), n_0)   cnf(outc$_0$, hypothesis)
$\neg$ mode(and$_1$(h), abnormal)   cnf(diagnosis_and$_1$, negated_conjecture)
$\neg$ mode(not$_1$(h), abnormal)   cnf(diagnosis_not$_1$, negated_conjecture)
$\neg$ mode(or$_1$(h), abnormal)   cnf(diagnosis_or$_1$, negated_conjecture)

HWV005-2.p A halfadder built from and, or and not gates
include('Axioms/HWV002-0.ax')
include('Axioms/HWV002-1.ax')
halfadder(h)   cnf(h_isa_halfadder, hypothesis)
1   cnf(in1$_1$, hypothesis)
0   cnf(in2$_0$, hypothesis)
0   cnf(outs$_0$, hypothesis)
0   cnf(outc$_0$, hypothesis)
$\neg$ abnormal(and$_1$(h))   cnf(diagnosis_and$_1$, negated_conjecture)
$\neg$ abnormal(not$_1$(h))   cnf(diagnosis_not$_1$, negated_conjecture)

$\neg\,\mathrm{abnormal}(\mathrm{or}_1(h))$ $\mathrm{cnf}(\mathrm{diagnosis\_or}_1, \mathrm{negated\_conjecture})$

HWV006-1.p A fulladder built from two halfadders and an or gate
include('Axioms/HWV001-0.ax')
include('Axioms/HWV001-1.ax')
include('Axioms/HWV001-2.ax')
$\mathrm{type}(f, \mathrm{fulladder})$ $\mathrm{cnf}(\mathrm{f\_isa\_fulladder}, \mathrm{hypothesis})$
$\mathrm{value}(\mathrm{in}(n_1, f), n_1)$ $\mathrm{cnf}(\mathrm{in1}_1, \mathrm{hypothesis})$
$\mathrm{value}(\mathrm{in}(n_2, f), n_0)$ $\mathrm{cnf}(\mathrm{in2}_0, \mathrm{hypothesis})$
$\mathrm{value}(\mathrm{in}(c, f), n_1)$ $\mathrm{cnf}(\mathrm{inc}_1, \mathrm{hypothesis})$
$\mathrm{value}(\mathrm{out}(s, f), n_1)$ $\mathrm{cnf}(\mathrm{outs}_1, \mathrm{hypothesis})$
$\mathrm{value}(\mathrm{out}(c, f), n_0)$ $\mathrm{cnf}(\mathrm{outc}_0, \mathrm{hypothesis})$
$\mathrm{mode}(\mathrm{or}_1(f), \mathrm{abnormal}) \Rightarrow \neg\,\mathrm{mode}(\mathrm{not}_1(h_2(f)), \mathrm{abnormal})$ $\mathrm{cnf}(\mathrm{diagnosis\_or1\_not1h}_2, \mathrm{negated\_conjecture})$
$\neg\,\mathrm{mode}(\mathrm{and}_2(h_2(f)), \mathrm{abnormal})$ $\mathrm{cnf}(\mathrm{diagnosis\_and}_2, \mathrm{negated\_conjecture})$
$\mathrm{mode}(\mathrm{or}_1(f), \mathrm{abnormal}) \Rightarrow \neg\,\mathrm{mode}(\mathrm{and}_1(h_2(f)), \mathrm{abnormal})$ $\mathrm{cnf}(\mathrm{diagnosis\_or1\_and1h}_2, \mathrm{negated\_conjecture})$
$\neg\,\mathrm{mode}(\mathrm{or}_1(h_1(f)), \mathrm{abnormal})$ $\mathrm{cnf}(\mathrm{diagnosis\_or1h}_1, \mathrm{negated\_conjecture})$
$\neg\,\mathrm{mode}(\mathrm{not}_1(h_1(f)), \mathrm{abnormal})$ $\mathrm{cnf}(\mathrm{diagnosis\_not1h}_1, \mathrm{negated\_conjecture})$
$\neg\,\mathrm{mode}(\mathrm{and}_2(h_1(f)), \mathrm{abnormal})$ $\mathrm{cnf}(\mathrm{diagnosis\_and2h}_1, \mathrm{negated\_conjecture})$
$\neg\,\mathrm{mode}(\mathrm{and}_1(h_1(f)), \mathrm{abnormal})$ $\mathrm{cnf}(\mathrm{diagnosis\_and1h}_1, \mathrm{negated\_conjecture})$

HWV006-2.p A fulladder built from two halfadders and an or gate
include('Axioms/HWV002-0.ax')
include('Axioms/HWV002-1.ax')
include('Axioms/HWV002-2.ax')
$\mathrm{fulladder}(f)$ $\mathrm{cnf}(\mathrm{f\_isa\_fulladder}, \mathrm{hypothesis})$
1 $\mathrm{cnf}(\mathrm{in1}_1, \mathrm{hypothesis})$
0 $\mathrm{cnf}(\mathrm{in2}_0, \mathrm{hypothesis})$
1 $\mathrm{cnf}(\mathrm{inc}_1, \mathrm{hypothesis})$
1 $\mathrm{cnf}(\mathrm{outs}_1, \mathrm{hypothesis})$
0 $\mathrm{cnf}(\mathrm{outc}_0, \mathrm{hypothesis})$
$\mathrm{abnormal}(\mathrm{or}_1(f)) \Rightarrow \neg\,\mathrm{abnormal}(\mathrm{not}_1(h_2(f)))$ $\mathrm{cnf}(\mathrm{diagnosis\_or1\_not1h}_2, \mathrm{negated\_conjecture})$
$\neg\,\mathrm{abnormal}(\mathrm{and}_2(h_2(f)))$ $\mathrm{cnf}(\mathrm{diagnosis\_and}_2, \mathrm{negated\_conjecture})$
$\mathrm{abnormal}(\mathrm{or}_1(f)) \Rightarrow \neg\,\mathrm{abnormal}(\mathrm{and}_1(h_2(f)))$ $\mathrm{cnf}(\mathrm{diagnosis\_or1\_and1h}_2, \mathrm{negated\_conjecture})$
$\neg\,\mathrm{abnormal}(\mathrm{or}_1(h_1(f)))$ $\mathrm{cnf}(\mathrm{diagnosis\_or1h}_1, \mathrm{negated\_conjecture})$
$\neg\,\mathrm{abnormal}(\mathrm{not}_1(h_1(f)))$ $\mathrm{cnf}(\mathrm{diagnosis\_not1h}_1, \mathrm{negated\_conjecture})$
$\neg\,\mathrm{abnormal}(\mathrm{and}_2(h_1(f)))$ $\mathrm{cnf}(\mathrm{diagnosis\_and2h}_1, \mathrm{negated\_conjecture})$
$\neg\,\mathrm{abnormal}(\mathrm{and}_1(h_1(f)))$ $\mathrm{cnf}(\mathrm{diagnosis\_and1h}_1, \mathrm{negated\_conjecture})$

HWV007-1.p A fulladder built from two halfadders and an or gate
include('Axioms/HWV001-0.ax')
include('Axioms/HWV001-1.ax')
include('Axioms/HWV001-2.ax')
$\mathrm{type}(f, \mathrm{fulladder})$ $\mathrm{cnf}(\mathrm{f\_isa\_fulladder}, \mathrm{hypothesis})$
$\mathrm{value}(\mathrm{in}(n_1, f), n_0)$ $\mathrm{cnf}(\mathrm{in1}_1, \mathrm{hypothesis})$
$\mathrm{value}(\mathrm{in}(n_2, f), n_1)$ $\mathrm{cnf}(\mathrm{in2}_0, \mathrm{hypothesis})$
$\mathrm{value}(\mathrm{in}(c, f), n_1)$ $\mathrm{cnf}(\mathrm{inc}_1, \mathrm{hypothesis})$
$\mathrm{value}(\mathrm{out}(s, f), n_1)$ $\mathrm{cnf}(\mathrm{outs}_1, \mathrm{hypothesis})$
$\mathrm{value}(\mathrm{out}(c, f), n_0)$ $\mathrm{cnf}(\mathrm{outc}_0, \mathrm{hypothesis})$
$\neg\,\mathrm{mode}(\mathrm{and}_2(h_1(f)), \mathrm{abnormal})$ $\mathrm{cnf}(\mathrm{diagnosis\_and2h}_1, \mathrm{negated\_conjecture})$
$\mathrm{mode}(\mathrm{or}_1(f), \mathrm{abnormal}) \Rightarrow \neg\,\mathrm{mode}(\mathrm{and}_1(h_1(f)), \mathrm{abnormal})$ $\mathrm{cnf}(\mathrm{diagnosis\_or1\_and1h}_1, \mathrm{negated\_conjecture})$
$\mathrm{mode}(\mathrm{or}_1(f), \mathrm{abnormal}) \Rightarrow \neg\,\mathrm{mode}(\mathrm{not}_1(h_1(f)), \mathrm{abnormal})$ $\mathrm{cnf}(\mathrm{diagnosis\_or1\_not1h}_1, \mathrm{negated\_conjecture})$
$\mathrm{mode}(\mathrm{or}_1(f), \mathrm{abnormal}) \Rightarrow \neg\,\mathrm{mode}(\mathrm{and}_1(h_2(f)), \mathrm{abnormal})$ $\mathrm{cnf}(\mathrm{diagnosis\_or1\_and1h}_2, \mathrm{negated\_conjecture})$
$\mathrm{mode}(\mathrm{or}_1(f), \mathrm{abnormal}) \Rightarrow \neg\,\mathrm{mode}(\mathrm{or}_1(h_2(f)), \mathrm{abnormal})$ $\mathrm{cnf}(\mathrm{diagnosis\_or1\_or1h}_2, \mathrm{negated\_conjecture})$

HWV007-2.p A fulladder built from two halfadders and an or gate
include('Axioms/HWV002-0.ax')
include('Axioms/HWV002-1.ax')
include('Axioms/HWV002-2.ax')
$\mathrm{fulladder}(f)$ $\mathrm{cnf}(\mathrm{f\_isa\_fulladder}, \mathrm{hypothesis})$
0 $\mathrm{cnf}(\mathrm{in1}_1, \mathrm{hypothesis})$
1 $\mathrm{cnf}(\mathrm{in2}_0, \mathrm{hypothesis})$

1 cnf(inc$_1$, hypothesis)
1 cnf(outs$_1$, hypothesis)
0 cnf(outc$_0$, hypothesis)
$\neg\,\mathrm{abnormal}(\mathrm{and}_2(h_1(f)))$ cnf(diagnosis_and2h$_1$, negated_conjecture)
$\mathrm{abnormal}(\mathrm{or}_1(f)) \Rightarrow \neg\,\mathrm{abnormal}(\mathrm{and}_1(h_1(f)))$ cnf(diagnosis_or1_and1h$_1$, negated_conjecture)
$\mathrm{abnormal}(\mathrm{or}_1(f)) \Rightarrow \neg\,\mathrm{abnormal}(\mathrm{not}_1(h_1(f)))$ cnf(diagnosis_or1_not1h$_1$, negated_conjecture)
$\mathrm{abnormal}(\mathrm{or}_1(f)) \Rightarrow \neg\,\mathrm{abnormal}(\mathrm{and}_1(h_2(f)))$ cnf(diagnosis_or1_and1h$_2$, negated_conjecture)
$\mathrm{abnormal}(\mathrm{or}_1(f)) \Rightarrow \neg\,\mathrm{abnormal}(\mathrm{or}_1(h_2(f)))$ cnf(diagnosis_or1_or1h$_2$, negated_conjecture)

HWV008-1.001.p 1 bit ripple carry adder
include('Axioms/HWV001-0.ax')
include('Axioms/HWV001-1.ax')
include('Axioms/HWV001-2.ax')
$\mathrm{type}(x, \mathrm{nbit\_adder}(n_1)) \Rightarrow \mathrm{type}(v(n_1, x), \mathrm{fulladder})$ cnf(nbit_adder_fulladder$_1$, axiom)
$\mathrm{type}(x, \mathrm{nbit\_adder}(n_1)) \Rightarrow \mathrm{connection}(\mathrm{out}(n_1, x), \mathrm{out}(s, v(n_1, x)))$ cnf(nbit_adder_connection_out1_out1v$_1$, axiom)
$\mathrm{type}(x, \mathrm{nbit\_adder}(n_1)) \Rightarrow \mathrm{connection}(\mathrm{out}(c, x), \mathrm{out}(c, v(n_1, x)))$ cnf(nbit_adder_connection_outc_outcv$_1$, axiom)
$\mathrm{type}(x, \mathrm{nbit\_adder}(n_1)) \Rightarrow \mathrm{connection}(\mathrm{in}(a_1, x), \mathrm{in}(n_1, v(n_1, x)))$ cnf(nbit_adder_connection_ina1_in1v$_1$, axiom)
$\mathrm{type}(x, \mathrm{nbit\_adder}(n_1)) \Rightarrow \mathrm{connection}(\mathrm{in}(b_1, x), \mathrm{in}(n_2, v(n_1, x)))$ cnf(nbit_adder_connection_inb1_in2v$_1$, axiom)
$\mathrm{type}(x, \mathrm{nbit\_adder}(n_1)) \Rightarrow \mathrm{connection}(\mathrm{in}(c, x), \mathrm{in}(c, v(n_1, x)))$ cnf(nbit_adder_connection_inc_incv$_1$, axiom)
$\mathrm{type}(a, \mathrm{nbit\_adder}(n_1))$ cnf(a_isa_1bit_adder, hypothesis)
$\mathrm{value}(\mathrm{in}(a_1, a), n_0)$ cnf(ina1$_0$, hypothesis)
$\mathrm{value}(\mathrm{in}(b_1, a), n_0)$ cnf(inb1$_0$, hypothesis)
$\mathrm{value}(\mathrm{in}(c, a), n_0)$ cnf(inc$_0$, hypothesis)
$\mathrm{value}(\mathrm{out}(n_1, a), n_0)$ cnf(out1$_0$, hypothesis)
$\mathrm{value}(\mathrm{out}(c, a), n_1)$ cnf(outc$_0$, hypothesis)
$\neg\,\mathrm{mode}(\mathrm{or}_1(v(n_1, a)), \mathrm{abnormal})$ cnf(diagnosis_or1v$_1$, negated_conjecture)
$\neg\,\mathrm{mode}(\mathrm{and}_2(h_1(v(n_1, a))), \mathrm{abnormal})$ cnf(diagnosis_and2h1v$_1$, negated_conjecture)
$\neg\,\mathrm{mode}(\mathrm{and}_2(h_2(v(n_1, a))), \mathrm{abnormal})$ cnf(diagnosis_and2h2v$_1$, negated_conjecture)

HWV008-1.002.p 2 bit ripple carry adder
include('Axioms/HWV001-0.ax')
include('Axioms/HWV001-1.ax')
include('Axioms/HWV001-2.ax')
$\mathrm{type}(x, \mathrm{nbit\_adder}(n_2)) \Rightarrow \mathrm{type}(f(n_1, x), \mathrm{fulladder})$ cnf(nbit_adder_fulladder$_1$, axiom)
$\mathrm{type}(x, \mathrm{nbit\_adder}(n_2)) \Rightarrow \mathrm{type}(f(n_2, x), \mathrm{fulladder})$ cnf(nbit_adder_fulladder$_2$, axiom)
$\mathrm{type}(x, \mathrm{nbit\_adder}(n_2)) \Rightarrow \mathrm{connection}(\mathrm{out}(n_1, x), \mathrm{out}(n_1, f(n_1, x)))$ cnf(nbit_adder_connection_out1_out1f$_1$, axiom)
$\mathrm{type}(x, \mathrm{nbit\_adder}(n_2)) \Rightarrow \mathrm{connection}(\mathrm{out}(n_2, x), \mathrm{out}(n_1, f(n_2, x)))$ cnf(nbit_adder_connection_out2_out1f$_2$, axiom)
$\mathrm{type}(x, \mathrm{nbit\_adder}(n_2)) \Rightarrow \mathrm{connection}(\mathrm{out}(c, x), \mathrm{out}(c, f(n_2, x)))$ cnf(nbit_adder_connection_outc_outcf$_1$, axiom)
$\mathrm{type}(x, \mathrm{nbit\_adder}(n_2)) \Rightarrow \mathrm{connection}(\mathrm{in}(a_1, x), \mathrm{in}(n_1, f(n_1, x)))$ cnf(nbit_adder_connection_ina1_in1f$_1$, axiom)
$\mathrm{type}(x, \mathrm{nbit\_adder}(n_2)) \Rightarrow \mathrm{connection}(\mathrm{in}(b_1, x), \mathrm{in}(n_2, f(n_1, x)))$ cnf(nbit_adder_connection_inb1_in2f$_1$, axiom)
$\mathrm{type}(x, \mathrm{nbit\_adder}(n_2)) \Rightarrow \mathrm{connection}(\mathrm{in}(a_2, x), \mathrm{in}(n_1, f(n_2, x)))$ cnf(nbit_adder_connection_ina2_in1f$_2$, axiom)
$\mathrm{type}(x, \mathrm{nbit\_adder}(n_2)) \Rightarrow \mathrm{connection}(\mathrm{in}(b_2, x), \mathrm{in}(n_2, f(n_2, x)))$ cnf(nbit_adder_connection_inb2_in2f$_2$, axiom)
$\mathrm{type}(x, \mathrm{nbit\_adder}(n_2)) \Rightarrow \mathrm{connection}(\mathrm{out}(c, f(n_1, x)), \mathrm{in}(c, f(n_2, x)))$ cnf(nbit_adder_connection_inc_incf$_1$, axiom)
$\mathrm{type}(x, \mathrm{nbit\_adder}(n_2)) \Rightarrow \mathrm{connection}(\mathrm{in}(c, x), \mathrm{in}(c, f(n_1, x)))$ cnf(nbit_adder_connection_inc_incf$_2$, axiom)
$\mathrm{type}(a, \mathrm{nbit\_adder}(n_2))$ cnf(a_isa_2bit_adder, hypothesis)
$\mathrm{value}(\mathrm{in}(a_1, a), n_0)$ cnf(ina1$_0$, hypothesis)
$\mathrm{value}(\mathrm{in}(a_2, a), n_0)$ cnf(ina2$_0$, hypothesis)
$\mathrm{value}(\mathrm{in}(b_1, a), n_0)$ cnf(inb1$_0$, hypothesis)
$\mathrm{value}(\mathrm{in}(b_2, a), n_0)$ cnf(inb2$_0$, hypothesis)
$\mathrm{value}(\mathrm{in}(c, a), n_0)$ cnf(inc$_0$, hypothesis)
$\mathrm{value}(\mathrm{out}(n_1, a), n_0)$ cnf(out1$_0$, hypothesis)
$\mathrm{value}(\mathrm{out}(n_2, a), n_0)$ cnf(out2$_0$, hypothesis)
$\mathrm{value}(\mathrm{out}(c, a), n_1)$ cnf(outc$_1$, hypothesis)
$\neg\,\mathrm{mode}(\mathrm{or}_1(f(n_2, a)), \mathrm{abnormal})$ cnf(diagnosis_or1f2a, negated_conjecture)
$\neg\,\mathrm{mode}(\mathrm{and}_2(h_1(f(n_2, a))), \mathrm{abnormal})$ cnf(diagnosis_and2h2f2a, negated_conjecture)
$\neg\,\mathrm{mode}(\mathrm{and}_2(h_2(f(n_2, a))), \mathrm{abnormal})$ cnf(diagnosis_and2h1f2a, negated_conjecture)

HWV008-2.001.p 1 bit ripple carry adder
include('Axioms/HWV002-0.ax')
include('Axioms/HWV002-1.ax')
include('Axioms/HWV002-2.ax')

$\mathrm{nbit\_adder}_1(x) \;\Rightarrow\; \mathrm{fulladder}(f_1(x))$ cnf(nbit_adder_fulladder$_1$, axiom)
$\mathrm{nbit\_adder}_1(x) \;\Rightarrow\; \mathrm{connection}(\mathrm{out}_1(x), \mathrm{outs}(f_1(x)))$ cnf(nbit_adder_connection_out1_out1f$_1$, axiom)
$\mathrm{nbit\_adder}_1(x) \;\Rightarrow\; \mathrm{connection}(\mathrm{outc}(x), \mathrm{outc}(f_1(x)))$ cnf(nbit_adder_connection_outc_outcf$_1$, axiom)
$\mathrm{nbit\_adder}_1(x) \;\Rightarrow\; \mathrm{connection}(\mathrm{ina}_1(x), \mathrm{in}_1(f_1(x)))$ cnf(nbit_adder_connection_ina1_in1f$_1$, axiom)
$\mathrm{nbit\_adder}_1(x) \;\Rightarrow\; \mathrm{connection}(\mathrm{inb}_1(x), \mathrm{in}_2(f_1(x)))$ cnf(nbit_adder_connection_inb1_in2f$_1$, axiom)
$\mathrm{nbit\_adder}_1(x) \;\Rightarrow\; \mathrm{connection}(\mathrm{inc}(x), \mathrm{inc}(f_1(x)))$ cnf(nbit_adder_connection_inc_incf$_1$, axiom)
$\mathrm{nbit\_adder}_1(a)$ cnf(a_isa_1bit_adder, hypothesis)
0 cnf(ina1$_0$, hypothesis)
0 cnf(inb1$_0$, hypothesis)
0 cnf(inc$_0$, hypothesis)
0 cnf(out1$_0$, hypothesis)
1 cnf(outc$_0$, hypothesis)
$\neg\, \mathrm{abnormal}(\mathrm{or}_1(f_1(a)))$ cnf(diagnosis_or1f$_1$, negated_conjecture)
$\neg\, \mathrm{abnormal}(\mathrm{and}_2(h_1(f_1(a))))$ cnf(diagnosis_and2h1f$_1$, negated_conjecture)
$\neg\, \mathrm{abnormal}(\mathrm{and}_2(h_2(f_1(a))))$ cnf(diagnosis_and2h2f$_1$, negated_conjecture)

HWV008-2.002.p 2 bit ripple carry adder
include('Axioms/HWV002-0.ax')
include('Axioms/HWV002-1.ax')
include('Axioms/HWV002-2.ax')
$\mathrm{nbit\_adder}_2(x) \;\Rightarrow\; \mathrm{fulladder}(f_1(x))$ cnf(nbit_adder_fulladder$_1$, axiom)
$\mathrm{nbit\_adder}_2(x) \;\Rightarrow\; \mathrm{fulladder}(f_2(x))$ cnf(nbit_adder_fulladder$_2$, axiom)
$\mathrm{nbit\_adder}_2(x) \;\Rightarrow\; \mathrm{connection}(\mathrm{out}_1(x), \mathrm{out}_1(f_1(x)))$ cnf(nbit_adder_connection_out1_out1f$_1$, axiom)
$\mathrm{nbit\_adder}_2(x) \;\Rightarrow\; \mathrm{connection}(\mathrm{out}_2(x), \mathrm{out}_1(f_2(x)))$ cnf(nbit_adder_connection_out2_out1f$_2$, axiom)
$\mathrm{nbit\_adder}_2(x) \;\Rightarrow\; \mathrm{connection}(\mathrm{outc}(x), \mathrm{outc}(f_2(x)))$ cnf(nbit_adder_connection_outc_outcf$_1$, axiom)
$\mathrm{nbit\_adder}_2(x) \;\Rightarrow\; \mathrm{connection}(\mathrm{ina}_1(x), \mathrm{in}_1(f_1(x)))$ cnf(nbit_adder_connection_ina1_in1f$_1$, axiom)
$\mathrm{nbit\_adder}_2(x) \;\Rightarrow\; \mathrm{connection}(\mathrm{inb}_1(x), \mathrm{in}_2(f_1(x)))$ cnf(nbit_adder_connection_inb1_in2f$_1$, axiom)
$\mathrm{nbit\_adder}_2(x) \;\Rightarrow\; \mathrm{connection}(\mathrm{ina}_2(x), \mathrm{in}_1(f_2(x)))$ cnf(nbit_adder_connection_ina2_in1f$_2$, axiom)
$\mathrm{nbit\_adder}_2(x) \;\Rightarrow\; \mathrm{connection}(\mathrm{inb}_2(x), \mathrm{in}_2(f_2(x)))$ cnf(nbit_adder_connection_inb2_in2f$_2$, axiom)
$\mathrm{nbit\_adder}_2(x) \;\Rightarrow\; \mathrm{connection}(\mathrm{outc}(f_1(x)), \mathrm{inc}(f_2(x)))$ cnf(nbit_adder_connection_inc_incf$_1$, axiom)
$\mathrm{nbit\_adder}_2(x) \;\Rightarrow\; \mathrm{connection}(\mathrm{inc}(x), \mathrm{inc}(f_1(x)))$ cnf(nbit_adder_connection_inc_incf$_2$, axiom)
$\mathrm{nbit\_adder}_2(a)$ cnf(a_isa_2bit_adder, hypothesis)
0 cnf(ina1$_0$, hypothesis)
0 cnf(ina2$_0$, hypothesis)
0 cnf(inb1$_0$, hypothesis)
0 cnf(inb2$_0$, hypothesis)
0 cnf(inc$_0$, hypothesis)
0 cnf(out1$_0$, hypothesis)
0 cnf(out2$_0$, hypothesis)
1 cnf(outc$_1$, hypothesis)
$\neg\, \mathrm{abnormal}(\mathrm{or}_1(f_2(a)))$ cnf(diagnosis_or1f2a, negated_conjecture)
$\neg\, \mathrm{abnormal}(\mathrm{and}_2(h_1(f_2(a))))$ cnf(diagnosis_and2h1f2a, negated_conjecture)
$\neg\, \mathrm{abnormal}(\mathrm{and}_2(h_2(f_2(a))))$ cnf(diagnosis_and2h2f2a, negated_conjecture)

HWV009-1.p Safelogic VHDL design verification obligation
include('Axioms/HWV003-0.ax')
$\mathrm{p\_Reset}(t_{139})$ cnf(quest$_1$, negated_conjecture)
$\mathrm{level}(t_{139} + n_1) = n_0 \;\Rightarrow\; (\mathrm{p\_Wr\_error}(t_{139} + n_1) \text{ or } \mathrm{p\_Rd\_error}(t_{139} + n_1))$ cnf(quest$_2$, negated_conjecture)

HWV009-2.p Safelogic VHDL design verification obligation
include('Axioms/HWV004-0.ax')
$\mathrm{p\_\_pred\_}(\mathrm{fwork\_DOTfifo\_DOTrtl\_DOTreset\_}(t_{206}))$ cnf(quest$_1$, negated_conjecture)
$\mathrm{fwork\_DOTfifo\_DOTrtl\_DOTlevel\_}(\mathrm{f\_ADD\_}(t_{206}, n_1)) \neq n_0$ cnf(quest$_2$, negated_conjecture)

HWV009-3.p Safelogic VHDL design verification obligation
include('Axioms/HWV004-0.ax')
$\mathrm{p\_\_pred\_}(\mathrm{fwork\_DOTfifo\_DOTrtl\_DOTreset\_}(t_{206}))$ cnf(quest$_1$, negated_conjecture)
$\mathrm{p\_\_pred\_}(\mathrm{fwork\_DOTfifo\_DOTrtl\_DOTwr\_\_error\_}(\mathrm{f\_ADD\_}(t_{206}, n_1)))$ cnf(quest$_2$, negated_conjecture)

HWV009-4.p Safelogic VHDL design verification obligation
include('Axioms/HWV004-0.ax')
$\mathrm{p\_\_pred\_}(\mathrm{fwork\_DOTfifo\_DOTrtl\_DOTreset\_}(t_{206}))$ cnf(quest$_1$, negated_conjecture)

p__pred_(fwork_DOTfifo_DOTrtl_DOTrd__error_(f_ADD_(t_{206}, n_1)))   cnf(quest$_2$, negated_conjecture)

HWV010-1.p Safelogic VHDL design verification obligation
include('Axioms/HWV003-0.ax')
p_Empty(t_{139}) $\Rightarrow$ level(t_{139}) $\neq n_0$ cnf(quest$_1$, negated_conjecture)
p_Empty(t_{139}) or level(t_{139}) $= n_0$ cnf(quest$_2$, negated_conjecture)

HWV010-2.p Safelogic VHDL design verification obligation
include('Axioms/HWV004-0.ax')
fwork_DOTfifo_DOTrtl_DOTlevel_(t_{206}) $\neq n_0$ cnf(quest$_1$, negated_conjecture)
p__pred_(fwork_DOTfifo_DOTrtl_DOTempty_(t_{206}))   cnf(quest$_2$, negated_conjecture)

HWV010-3.p Safelogic VHDL design verification obligation
include('Axioms/HWV004-0.ax')
fwork_DOTfifo_DOTrtl_DOTlevel_(t_{206}) $= n_0$ cnf(quest$_1$, negated_conjecture)
$\neg$ p__pred_(fwork_DOTfifo_DOTrtl_DOTempty_(t_{206}))   cnf(quest$_2$, negated_conjecture)

HWV011-1.p Safelogic VHDL design verification obligation
include('Axioms/HWV003-0.ax')
p_Full(t_{139}) $\Rightarrow$ level(t_{139}) $\neq$ fifo_length   cnf(quest$_1$, negated_conjecture)
p_Full(t_{139}) or level(t_{139}) = fifo_length   cnf(quest$_2$, negated_conjecture)

HWV011-2.p Safelogic VHDL design verification obligation
include('Axioms/HWV004-0.ax')
fwork_DOTfifo_DOTrtl_DOTlevel_(t_{206}) $\neq$ fwork_DOTfifo_DOTrtl_DOTfifo__length_   cnf(quest$_1$, negated_conjecture)
p__pred_(fwork_DOTfifo_DOTrtl_DOTfull_(t_{206}))   cnf(quest$_2$, negated_conjecture)

HWV011-3.p Safelogic VHDL design verification obligation
include('Axioms/HWV004-0.ax')
fwork_DOTfifo_DOTrtl_DOTlevel_(t_{206}) = fwork_DOTfifo_DOTrtl_DOTfifo__length_   cnf(quest$_1$, negated_conjecture)
$\neg$ p__pred_(fwork_DOTfifo_DOTrtl_DOTfull_(t_{206}))   cnf(quest$_2$, negated_conjecture)

HWV012-1.p Safelogic VHDL design verification obligation
include('Axioms/HWV003-0.ax')
gt(fifo_length, level(t_{139}))   cnf(quest$_1$, negated_conjecture)
p_Wr(t_{139})   cnf(quest$_2$, negated_conjecture)
p_Wr_error($t_{139} + n_1$)   cnf(quest$_3$, negated_conjecture)

HWV012-2.p Safelogic VHDL design verification obligation
include('Axioms/HWV004-0.ax')
$\neg$ p_LES_EQU_(fwork_DOTfifo_DOTrtl_DOTfifo__length_, fwork_DOTfifo_DOTrtl_DOTlevel_(t_{206}))   cnf(quest$_1$, negated_co
p__pred_(fwork_DOTfifo_DOTrtl_DOTwr_(t_{206}))   cnf(quest$_2$, negated_conjecture)
p__pred_(fwork_DOTfifo_DOTrtl_DOTwr__error_(f_ADD_(t_{206}, n_1)))   cnf(quest$_3$, negated_conjecture)

HWV013-1.p Safelogic VHDL design verification obligation
include('Axioms/HWV003-0.ax')
$\neg$ p_Reset(x_{139})   cnf(quest$_1$, negated_conjecture)
gt(fifo_length, level(x_{139}))   cnf(quest$_2$, negated_conjecture)
$\neg$ p_Rd(x_{139})   cnf(quest$_3$, negated_conjecture)
p_Wr(x_{139})   cnf(quest$_4$, negated_conjecture)
level($x_{139} + n_1$) $\neq$ level(x_{139}) $+ n_1$ cnf(quest$_5$, negated_conjecture)

HWV013-2.p Safelogic VHDL design verification obligation
include('Axioms/HWV004-0.ax')
$\neg$ p__pred_(fwork_DOTfifo_DOTrtl_DOTreset_(t_{206}))   cnf(quest$_1$, negated_conjecture)
$\neg$ p__pred_(fwork_DOTfifo_DOTrtl_DOTrd_(t_{206}))   cnf(quest$_2$, negated_conjecture)
p__pred_(fwork_DOTfifo_DOTrtl_DOTwr_(t_{206}))   cnf(quest$_3$, negated_conjecture)
$\neg$ p_LES_EQU_(fwork_DOTfifo_DOTrtl_DOTfifo__length_, fwork_DOTfifo_DOTrtl_DOTlevel_(t_{206}))   cnf(quest$_4$, negated_co
fwork_DOTfifo_DOTrtl_DOTlevel_(f_ADD_(t_{206}, n_1)) $\neq$ f_ADD_(fwork_DOTfifo_DOTrtl_DOTlevel_(t_{206}), n_1)   cnf(quest$_5$, r

HWV014-1.p Safelogic VHDL design verification obligation
include('Axioms/HWV003-0.ax')
gt(level(x_{139}), n_0)   cnf(quest$_1$, negated_conjecture)
p_Rd(x_{139})   cnf(quest$_2$, negated_conjecture)
$\neg$ p_Wr(x_{139})   cnf(quest$_3$, negated_conjecture)
$\neg$ p_Reset(x_{139})   cnf(quest$_4$, negated_conjecture)

level($x_{139} + n_1$) $\neq$ −level(x_{139})   cnf(quest$_5$, negated_conjecture)

HWV014-2.p Safelogic VHDL design verification obligation
include('Axioms/HWV004-0.ax')
¬ p_LES_EQU_(fwork_DOTfifo_DOTrtl_DOTlevel_(t_{206}), n_0)   cnf(quest$_1$, negated_conjecture)
¬ p__pred_(fwork_DOTfifo_DOTrtl_DOTwr_(t_{206}))   cnf(quest$_2$, negated_conjecture)
¬ p__pred_(fwork_DOTfifo_DOTrtl_DOTreset_(t_{206}))   cnf(quest$_3$, negated_conjecture)
p__pred_(fwork_DOTfifo_DOTrtl_DOTrd_(t_{206}))   cnf(quest$_4$, negated_conjecture)
fwork_DOTfifo_DOTrtl_DOTlevel_(f_ADD_(t_{206}, n_1)) $\neq$ f_SUB_(fwork_DOTfifo_DOTrtl_DOTlevel_(t_{206}), n_1)   cnf(quest$_5$, n

HWV015-1.p Safelogic VHDL design verification obligation
include('Axioms/HWV003-0.ax')
gt(level(x_{139}), n_0)   cnf(quest$_1$, negated_conjecture)
p_Rd(x_{139})   cnf(quest$_2$, negated_conjecture)
p_Wr(x_{139})   cnf(quest$_3$, negated_conjecture)
¬ p_Reset(x_{139})   cnf(quest$_4$, negated_conjecture)
level($x_{139} + n_1$) $\neq$ level(x_{139})   cnf(quest$_5$, negated_conjecture)

HWV015-2.p Safelogic VHDL design verification obligation
include('Axioms/HWV004-0.ax')
¬ p_LES_EQU_(fwork_DOTfifo_DOTrtl_DOTlevel_(t_{206}), n_0)   cnf(quest$_1$, negated_conjecture)
p__pred_(fwork_DOTfifo_DOTrtl_DOTwr_(t_{206}))   cnf(quest$_2$, negated_conjecture)
¬ p__pred_(fwork_DOTfifo_DOTrtl_DOTreset_(t_{206}))   cnf(quest$_3$, negated_conjecture)
p__pred_(fwork_DOTfifo_DOTrtl_DOTrd_(t_{206}))   cnf(quest$_4$, negated_conjecture)
fwork_DOTfifo_DOTrtl_DOTlevel_(f_ADD_(t_{206}, n_1)) $\neq$ fwork_DOTfifo_DOTrtl_DOTlevel_(t_{206})   cnf(quest$_5$, negated_conje

HWV016-1.p Safelogic VHDL design verification obligation
include('Axioms/HWV003-0.ax')
¬ gt(n_0, level(t_{139}))   cnf(quest$_1$, negated_conjecture)
p_Wr(t_{139})   cnf(quest$_2$, negated_conjecture)
¬ p_Reset(t_{139})   cnf(quest$_3$, negated_conjecture)
p_Empty($t_{139} + n_1$)   cnf(quest$_4$, negated_conjecture)
gt(fifo_length, n_0)   cnf(quest$_5$, negated_conjecture)

HWV016-2.p Safelogic VHDL design verification obligation
include('Axioms/HWV004-0.ax')
p_LES_EQU_(n_0, fwork_DOTfifo_DOTrtl_DOTlevel_(t_{206}))   cnf(quest$_1$, negated_conjecture)
¬ p__pred_(fwork_DOTfifo_DOTrtl_DOTreset_(t_{206}))   cnf(quest$_2$, negated_conjecture)
p__pred_(fwork_DOTfifo_DOTrtl_DOTwr_(t_{206}))   cnf(quest$_3$, negated_conjecture)
p__pred_(fwork_DOTfifo_DOTrtl_DOTempty_(f_ADD_(t_{206}, n_1)))   cnf(quest$_4$, negated_conjecture)
¬ p_LES_EQU_(fwork_DOTfifo_DOTrtl_DOTfifo__length_, n_0)   cnf(quest$_5$, negated_conjecture)

HWV017-1.p Safelogic VHDL design verification obligation
include('Axioms/HWV003-0.ax')
¬ gt(level(t_{139}), fifo_length)   cnf(quest$_1$, negated_conjecture)
p_Rd(t_{139})   cnf(quest$_2$, negated_conjecture)
¬ p_Wr(t_{139})   cnf(quest$_3$, negated_conjecture)
¬ p_Reset(t_{139})   cnf(quest$_4$, negated_conjecture)
p_Full($t_{139} + n_1$)   cnf(quest$_5$, negated_conjecture)
gt(fifo_length, n_0)   cnf(quest$_6$, negated_conjecture)

HWV017-2.p Safelogic VHDL design verification obligation
include('Axioms/HWV004-0.ax')
p_LES_EQU_(fwork_DOTfifo_DOTrtl_DOTlevel_(t_{206}), fwork_DOTfifo_DOTrtl_DOTfifo__length_)   cnf(quest$_1$, negated_con
¬ p__pred_(fwork_DOTfifo_DOTrtl_DOTwr_(t_{206}))   cnf(quest$_2$, negated_conjecture)
¬ p__pred_(fwork_DOTfifo_DOTrtl_DOTreset_(t_{206}))   cnf(quest$_3$, negated_conjecture)
p__pred_(fwork_DOTfifo_DOTrtl_DOTrd_(t_{206}))   cnf(quest$_4$, negated_conjecture)
p__pred_(fwork_DOTfifo_DOTrtl_DOTfull_(f_ADD_(t_{206}, n_1)))   cnf(quest$_5$, negated_conjecture)
¬ p_LES_EQU_(fwork_DOTfifo_DOTrtl_DOTfifo__length_, n_0)   cnf(quest$_6$, negated_conjecture)

HWV018-1.p Safelogic VHDL design verification obligation
include('Axioms/HWV003-0.ax')
¬ p_Rd(t_{139})   cnf(quest$_1$, negated_conjecture)
p_Wr(t_{139})   cnf(quest$_2$, negated_conjecture)

p_Full(t_{139}) cnf(quest$_3$, negated_conjecture)
$\neg$ p_Reset(t_{139}) cnf(quest$_4$, negated_conjecture)
p_Full($t_{139} + n_1$) $\Rightarrow$ $\neg$ p_Wr_error($t_{139} + n_1$) cnf(quest$_5$, negated_conjecture)

HWV018-2.p Safelogic VHDL design verification obligation
include('Axioms/HWV004-0.ax')
$\neg$ p__pred_(fwork_DOTfifo_DOTrtl_DOTrd_(t_{206})) cnf(quest$_1$, negated_conjecture)
p__pred_(fwork_DOTfifo_DOTrtl_DOTfull_(t_{206})) cnf(quest$_2$, negated_conjecture)
$\neg$ p__pred_(fwork_DOTfifo_DOTrtl_DOTreset_(t_{206})) cnf(quest$_3$, negated_conjecture)
p__pred_(fwork_DOTfifo_DOTrtl_DOTwr_(t_{206})) cnf(quest$_4$, negated_conjecture)
$\neg$ p__pred_(fwork_DOTfifo_DOTrtl_DOTfull_(f_ADD_(t_{206}, n_1))) cnf(quest$_5$, negated_conjecture)

HWV018-3.p Safelogic VHDL design verification obligation
include('Axioms/HWV004-0.ax')
$\neg$ p__pred_(fwork_DOTfifo_DOTrtl_DOTrd_(t_{206})) cnf(quest$_1$, negated_conjecture)
p__pred_(fwork_DOTfifo_DOTrtl_DOTfull_(t_{206})) cnf(quest$_2$, negated_conjecture)
$\neg$ p__pred_(fwork_DOTfifo_DOTrtl_DOTreset_(t_{206})) cnf(quest$_3$, negated_conjecture)
p__pred_(fwork_DOTfifo_DOTrtl_DOTwr_(t_{206})) cnf(quest$_4$, negated_conjecture)
$\neg$ p__pred_(fwork_DOTfifo_DOTrtl_DOTwr__error_(f_ADD_(t_{206}, n_1))) cnf(quest$_5$, negated_conjecture)

HWV019-1.p Safelogic VHDL design verification obligation
include('Axioms/HWV003-0.ax')
p_Wr_error($t_{139} + n_1$) cnf(quest$_1$, negated_conjecture)
p_Wr(t_{139}) cnf(quest$_2$, negated_conjecture)
$\neg$ p_Reset(t_{139}) cnf(quest$_3$, negated_conjecture)
gt(fifo_length, level(t_{139})) or p_Rd(t_{139}) cnf(quest$_4$, negated_conjecture)

HWV019-2.p Safelogic VHDL design verification obligation
include('Axioms/HWV004-0.ax')
p__pred_(fwork_DOTfifo_DOTrtl_DOTwr__error_(f_ADD_(t_{206}, n_1))) cnf(quest$_1$, negated_conjecture)
$\neg$ p__pred_(fwork_DOTfifo_DOTrtl_DOTreset_(t_{206})) cnf(quest$_2$, negated_conjecture)
p__pred_(fwork_DOTfifo_DOTrtl_DOTwr_(t_{206})) cnf(quest$_3$, negated_conjecture)
$\neg$ p_LES_EQU_(fwork_DOTfifo_DOTrtl_DOTfifo__length_, fwork_DOTfifo_DOTrtl_DOTlevel_(t_{206})) cnf(quest$_4$, negated_co

HWV019-3.p Safelogic VHDL design verification obligation
include('Axioms/HWV004-0.ax')
p__pred_(fwork_DOTfifo_DOTrtl_DOTwr__error_(f_ADD_(t_{206}, n_1))) cnf(quest$_1$, negated_conjecture)
$\neg$ p__pred_(fwork_DOTfifo_DOTrtl_DOTreset_(t_{206})) cnf(quest$_2$, negated_conjecture)
p__pred_(fwork_DOTfifo_DOTrtl_DOTwr_(t_{206})) cnf(quest$_3$, negated_conjecture)
p__pred_(fwork_DOTfifo_DOTrtl_DOTrd_(t_{206})) cnf(quest$_4$, negated_conjecture)

HWV020-1.p Safelogic VHDL design verification obligation
include('Axioms/HWV003-0.ax')
p_Rd(t_{139}) cnf(quest$_1$, negated_conjecture)
p_Empty(t_{139}) cnf(quest$_2$, negated_conjecture)
$\neg$ p_Wr(t_{139}) cnf(quest$_3$, negated_conjecture)
$\neg$ p_Reset(t_{139}) cnf(quest$_4$, negated_conjecture)
p_Empty($t_{139} + n_1$) $\Rightarrow$ $\neg$ p_Rd_error($t_{139} + n_1$) cnf(quest$_5$, negated_conjecture)

HWV020-2.p Safelogic VHDL design verification obligation
include('Axioms/HWV004-0.ax')
p__pred_(fwork_DOTfifo_DOTrtl_DOTrd_(t_{206})) cnf(quest$_1$, negated_conjecture)
$\neg$ p__pred_(fwork_DOTfifo_DOTrtl_DOTwr_(t_{206})) cnf(quest$_2$, negated_conjecture)
$\neg$ p__pred_(fwork_DOTfifo_DOTrtl_DOTreset_(t_{206})) cnf(quest$_3$, negated_conjecture)
p__pred_(fwork_DOTfifo_DOTrtl_DOTempty_(t_{206})) cnf(quest$_4$, negated_conjecture)
$\neg$ p__pred_(fwork_DOTfifo_DOTrtl_DOTempty_(f_ADD_(t_{206}, n_1))) cnf(quest$_5$, negated_conjecture)

HWV020-3.p Safelogic VHDL design verification obligation
include('Axioms/HWV004-0.ax')
p__pred_(fwork_DOTfifo_DOTrtl_DOTrd_(t_{206})) cnf(quest$_1$, negated_conjecture)
$\neg$ p__pred_(fwork_DOTfifo_DOTrtl_DOTwr_(t_{206})) cnf(quest$_2$, negated_conjecture)
$\neg$ p__pred_(fwork_DOTfifo_DOTrtl_DOTreset_(t_{206})) cnf(quest$_3$, negated_conjecture)
p__pred_(fwork_DOTfifo_DOTrtl_DOTempty_(t_{206})) cnf(quest$_4$, negated_conjecture)
$\neg$ p__pred_(fwork_DOTfifo_DOTrtl_DOTrd__error_(f_ADD_(t_{206}, n_1))) cnf(quest$_5$, negated_conjecture)

HWV021-1.p Safelogic VHDL design verification obligation
include('Axioms/HWV003-0.ax')
p_Rd_error($t_{139} + n_1$)   cnf(quest$_1$, negated_conjecture)
p_Rd(t_{139})   cnf(quest$_2$, negated_conjecture)
$\neg$ p_Reset(t_{139})   cnf(quest$_3$, negated_conjecture)
$\neg$ p_Empty(t_{139})   cnf(quest$_4$, negated_conjecture)

HWV021-2.p Safelogic VHDL design verification obligation
include('Axioms/HWV004-0.ax')
p__pred_(fwork_DOTfifo_DOTrtl_DOTrd__error_(f_ADD_(t_{206}, n_1)))   cnf(quest$_1$, negated_conjecture)
$\neg$ p__pred_(fwork_DOTfifo_DOTrtl_DOTreset_(t_{206}))   cnf(quest$_2$, negated_conjecture)
p__pred_(fwork_DOTfifo_DOTrtl_DOTrd_(t_{206}))   cnf(quest$_3$, negated_conjecture)
$\neg$ p__pred_(fwork_DOTfifo_DOTrtl_DOTempty_(t_{206}))   cnf(quest$_4$, negated_conjecture)

HWV022-1.p Safelogic VHDL design verification obligation
include('Axioms/HWV003-0.ax')
level(t_{139}) + n_1 = fifo_length   cnf(quest$_1$, negated_conjecture)
$\neg$ p_Rd(t_{139})   cnf(quest$_2$, negated_conjecture)
p_Wr(t_{139})   cnf(quest$_3$, negated_conjecture)
$\neg$ p_Reset(t_{139})   cnf(quest$_4$, negated_conjecture)
level($t_{139} + n_1$) $\neq$ fifo_length   cnf(quest$_5$, negated_conjecture)

HWV022-2.p Safelogic VHDL design verification obligation
include('Axioms/HWV004-0.ax')
f_ADD_(fwork_DOTfifo_DOTrtl_DOTlevel_(t_{206}), n_1) = fwork_DOTfifo_DOTrtl_DOTfifo__length_   cnf(quest$_1$, negated_con
p__pred_(fwork_DOTfifo_DOTrtl_DOTwr_(t_{206}))   cnf(quest$_2$, negated_conjecture)
$\neg$ p__pred_(fwork_DOTfifo_DOTrtl_DOTreset_(t_{206}))   cnf(quest$_3$, negated_conjecture)
$\neg$ p__pred_(fwork_DOTfifo_DOTrtl_DOTrd_(t_{206}))   cnf(quest$_4$, negated_conjecture)
fwork_DOTfifo_DOTrtl_DOTlevel_(f_ADD_(t_{206}, n_1)) $\neq$ fwork_DOTfifo_DOTrtl_DOTfifo__length_   cnf(quest$_5$, negated_con

HWV023-1.p Safelogic VHDL design verification obligation
include('Axioms/HWV003-0.ax')
level(t_{139}) = n_1 cnf(quest$_1$, negated_conjecture)
$\neg$ p_Wr(t_{139})   cnf(quest$_2$, negated_conjecture)
p_Rd(t_{139})   cnf(quest$_3$, negated_conjecture)
$\neg$ p_Reset(t_{139})   cnf(quest$_4$, negated_conjecture)
$\neg$ p_Empty($t_{139} + n_1$)   cnf(quest$_5$, negated_conjecture)

HWV023-2.p Safelogic VHDL design verification obligation
include('Axioms/HWV004-0.ax')
fwork_DOTfifo_DOTrtl_DOTlevel_(t_{206}) = n_1 cnf(quest$_1$, negated_conjecture)
p__pred_(fwork_DOTfifo_DOTrtl_DOTrd_(t_{206}))   cnf(quest$_2$, negated_conjecture)
$\neg$ p__pred_(fwork_DOTfifo_DOTrtl_DOTreset_(t_{206}))   cnf(quest$_3$, negated_conjecture)
$\neg$ p__pred_(fwork_DOTfifo_DOTrtl_DOTwr_(t_{206}))   cnf(quest$_4$, negated_conjecture)
$\neg$ p__pred_(fwork_DOTfifo_DOTrtl_DOTempty_(f_ADD_(t_{206}, n_1)))   cnf(quest$_5$, negated_conjecture)

HWV024-1.p Safelogic VHDL design verification obligation
include('Axioms/HWV003-0.ax')
$\neg$ p_Rd(x_{140})   cnf(quest$_1$, negated_conjecture)
$\neg$ p_Reset(x_{140})   cnf(quest$_2$, negated_conjecture)
p_Data_out(x_{139}, x_{140}) $\Rightarrow$ $\neg$ p_Data_out($x_{139}, x_{140} + n_1$)   cnf(quest$_3$, negated_conjecture)
p_Data_out(x_{139}, x_{140}) or p_Data_out($x_{139}, x_{140} + n_1$)   cnf(quest$_4$, negated_conjecture)

HWV024-2.p Safelogic VHDL design verification obligation
include('Axioms/HWV004-0.ax')
$\neg$ p__pred_(fwork_DOTfifo_DOTrtl_DOTrd_(t_{206}))   cnf(quest$_1$, negated_conjecture)
$\neg$ p__pred_(fwork_DOTfifo_DOTrtl_DOTreset_(t_{206}))   cnf(quest$_2$, negated_conjecture)
p_LES_EQU_(n_0, x_{207})   cnf(quest$_3$, negated_conjecture)
p_LES_EQU_(x_{207}, f_SUB_(fwork_DOTfifo_DOTrtl_DOTfifo__width_, n_1))   cnf(quest$_4$, negated_conjecture)
f__index_(fwork_DOTfifo_DOTrtl_DOTdata__out_(f_ADD_(t_{206}, n_1)), f_SUB_(f_SUB_(fwork_DOTfifo_DOTrtl_DOTfifo__width_,
f__index_(fwork_DOTfifo_DOTrtl_DOTdata__out_(t_{206}), f_SUB_(f_SUB_(fwork_DOTfifo_DOTrtl_DOTfifo__width_, n_1), x_{207}))

HWV025-1.p Safelogic VHDL design verification obligation
include('Axioms/HWV003-0.ax')

¬ p_Rd(t_{139})   cnf(quest$_1$, negated_conjecture)
¬ p_Reset(t_{139})   cnf(quest$_2$, negated_conjecture)
p_Rd_error(t_{139}) $\Rightarrow$ ¬ p_Rd_error($t_{139} + n_1$)   cnf(quest$_3$, negated_conjecture)
p_Rd_error(t_{139}) or p_Rd_error($t_{139} + n_1$)   cnf(quest$_4$, negated_conjecture)

HWV025-2.p Safelogic VHDL design verification obligation
include('Axioms/HWV004-0.ax')
¬ p__pred_(fwork_DOTfifo_DOTrtl_DOTrd_(t_{206}))   cnf(quest$_1$, negated_conjecture)
¬ p__pred_(fwork_DOTfifo_DOTrtl_DOTreset_(t_{206}))   cnf(quest$_2$, negated_conjecture)
¬ p__pred_(fwork_DOTfifo_DOTrtl_DOTrd__error_(f_ADD_(t_{206}, n_1)))   cnf(quest$_3$, negated_conjecture)
p__pred_(fwork_DOTfifo_DOTrtl_DOTrd__error_(t_{206}))   cnf(quest$_4$, negated_conjecture)

HWV025-3.p Safelogic VHDL design verification obligation
include('Axioms/HWV004-0.ax')
¬ p__pred_(fwork_DOTfifo_DOTrtl_DOTrd_(t_{206}))   cnf(quest$_1$, negated_conjecture)
¬ p__pred_(fwork_DOTfifo_DOTrtl_DOTreset_(t_{206}))   cnf(quest$_2$, negated_conjecture)
p__pred_(fwork_DOTfifo_DOTrtl_DOTrd__error_(f_ADD_(t_{206}, n_1)))   cnf(quest$_3$, negated_conjecture)
¬ p__pred_(fwork_DOTfifo_DOTrtl_DOTrd__error_(t_{206}))   cnf(quest$_4$, negated_conjecture)

HWV026-1.p Safelogic VHDL design verification obligation
include('Axioms/HWV003-0.ax')
¬ p_Wr(t_{139})   cnf(quest$_1$, negated_conjecture)
¬ p_Reset(t_{139})   cnf(quest$_2$, negated_conjecture)
p_Wr_error(t_{139}) $\Rightarrow$ ¬ p_Wr_error($t_{139} + n_1$)   cnf(quest$_3$, negated_conjecture)
p_Wr_error(t_{139}) or p_Wr_error($t_{139} + n_1$)   cnf(quest$_4$, negated_conjecture)

HWV026-2.p Safelogic VHDL design verification obligation
include('Axioms/HWV004-0.ax')
¬ p__pred_(fwork_DOTfifo_DOTrtl_DOTwr_(t_{206}))   cnf(quest$_1$, negated_conjecture)
¬ p__pred_(fwork_DOTfifo_DOTrtl_DOTreset_(t_{206}))   cnf(quest$_2$, negated_conjecture)
¬ p__pred_(fwork_DOTfifo_DOTrtl_DOTwr__error_(f_ADD_(t_{206}, n_1)))   cnf(quest$_3$, negated_conjecture)
p__pred_(fwork_DOTfifo_DOTrtl_DOTwr__error_(t_{206}))   cnf(quest$_4$, negated_conjecture)

HWV026-3.p Safelogic VHDL design verification obligation
include('Axioms/HWV004-0.ax')
¬ p__pred_(fwork_DOTfifo_DOTrtl_DOTwr_(t_{206}))   cnf(quest$_1$, negated_conjecture)
¬ p__pred_(fwork_DOTfifo_DOTrtl_DOTreset_(t_{206}))   cnf(quest$_2$, negated_conjecture)
p__pred_(fwork_DOTfifo_DOTrtl_DOTwr__error_(f_ADD_(t_{206}, n_1)))   cnf(quest$_3$, negated_conjecture)
¬ p__pred_(fwork_DOTfifo_DOTrtl_DOTwr__error_(t_{206}))   cnf(quest$_4$, negated_conjecture)

HWV027-1.p Safelogic VHDL design verification obligation
include('Axioms/HWV003-0.ax')
¬ p_Wr(t_{139})   cnf(quest$_1$, negated_conjecture)
¬ p_Rd(t_{139})   cnf(quest$_2$, negated_conjecture)
¬ p_Reset(t_{139})   cnf(quest$_3$, negated_conjecture)
level(t_{139}) $\neq$ level($t_{139} + n_1$)   cnf(quest$_4$, negated_conjecture)

HWV027-2.p Safelogic VHDL design verification obligation
include('Axioms/HWV004-0.ax')
¬ p__pred_(fwork_DOTfifo_DOTrtl_DOTwr_(t_{206}))   cnf(quest$_1$, negated_conjecture)
¬ p__pred_(fwork_DOTfifo_DOTrtl_DOTreset_(t_{206}))   cnf(quest$_2$, negated_conjecture)
¬ p__pred_(fwork_DOTfifo_DOTrtl_DOTrd_(t_{206}))   cnf(quest$_3$, negated_conjecture)
fwork_DOTfifo_DOTrtl_DOTlevel_(t_{206}) $\neq$ fwork_DOTfifo_DOTrtl_DOTlevel_(f_ADD_(t_{206}, n_1))   cnf(quest$_4$, negated_conje

HWV028-1.p Safelogic VHDL design verification obligation
include('Axioms/HWV003-0.ax')
p_Reset(t_{139})   cnf(quest$_1$, negated_conjecture)
gt(level($t_{139} + n_1$), fifo_length)   cnf(quest$_2$, negated_conjecture)
gt(fifo_length, n_0)   cnf(quest$_3$, negated_conjecture)

HWV028-2.p Safelogic VHDL design verification obligation
include('Axioms/HWV004-0.ax')
p__pred_(fwork_DOTfifo_DOTrtl_DOTreset_(t_{206}))   cnf(quest$_1$, negated_conjecture)
¬ p_LES_EQU_(fwork_DOTfifo_DOTrtl_DOTlevel_(f_ADD_(t_{206}, n_1)), fwork_DOTfifo_DOTrtl_DOTfifo__length_)   cnf(quest
¬ p_LES_EQU_(fwork_DOTfifo_DOTrtl_DOTfifo__length_, n_0)   cnf(quest$_3$, negated_conjecture)

HWV029-1.p Safelogic VHDL design verification obligation
include('Axioms/HWV003-0.ax')
¬ gt(level(x_{139}), fifo_length)   cnf(quest$_1$, negated_conjecture)
gt(level($x_{139} + n_1$), fifo_length)   cnf(quest$_2$, negated_conjecture)
gt(fifo_length, n_0)   cnf(quest$_3$, negated_conjecture)

HWV029-2.p Safelogic VHDL design verification obligation
include('Axioms/HWV004-0.ax')
p_LES_EQU_(fwork_DOTfifo_DOTrtl_DOTlevel_(t_{206}), fwork_DOTfifo_DOTrtl_DOTfifo__length_)   cnf(quest$_1$, negated_con
¬ p_LES_EQU_(fwork_DOTfifo_DOTrtl_DOTlevel_(f_ADD_(t_{206}, n_1)), fwork_DOTfifo_DOTrtl_DOTfifo__length_)   cnf(quest
¬ p_LES_EQU_(fwork_DOTfifo_DOTrtl_DOTfifo__length_, n_0)   cnf(quest$_3$, negated_conjecture)

HWV030-1.p Safelogic VHDL design verification obligation
include('Axioms/HWV003-0.ax')
p_Reset(t_{139})   cnf(quest$_1$, negated_conjecture)
¬ gt(fifo_length, wr_level($t_{139} + n_1$))   cnf(quest$_2$, negated_conjecture)
gt(fifo_length, n_0)   cnf(quest$_3$, negated_conjecture)

HWV030-2.p Safelogic VHDL design verification obligation
include('Axioms/HWV004-0.ax')
p__pred_(fwork_DOTfifo_DOTrtl_DOTreset_(t_{206}))   cnf(quest$_1$, negated_conjecture)
p_LES_EQU_(fwork_DOTfifo_DOTrtl_DOTfifo__length_, fwork_DOTfifo_DOTrtl_DOTwr__level_(f_ADD_(t_{206}, n_1)))   cnf(qu
¬ p_LES_EQU_(fwork_DOTfifo_DOTrtl_DOTfifo__length_, n_0)   cnf(quest$_3$, negated_conjecture)

HWV031-1.p Safelogic VHDL design verification obligation
include('Axioms/HWV003-0.ax')
¬ gt(wr_level(x_{139}) + n_1, fifo_length)   cnf(quest$_1$, negated_conjecture)
gt(wr_level($x_{139} + n_1$) + n_1, fifo_length)   cnf(quest$_2$, negated_conjecture)
gt(fifo_length, n_0)   cnf(quest$_3$, negated_conjecture)

HWV031-2.p Safelogic VHDL design verification obligation
include('Axioms/HWV004-0.ax')
p_LES_EQU_(f_ADD_(fwork_DOTfifo_DOTrtl_DOTwr__level_(t_{206}), n_1), fwork_DOTfifo_DOTrtl_DOTfifo__length_)   cnf(qu
¬ p_LES_EQU_(f_ADD_(fwork_DOTfifo_DOTrtl_DOTwr__level_(f_ADD_(t_{206}, n_1)), n_1), fwork_DOTfifo_DOTrtl_DOTfifo__leng
¬ p_LES_EQU_(fwork_DOTfifo_DOTrtl_DOTfifo__length_, n_0)   cnf(quest$_3$, negated_conjecture)

HWV032-1.p Safelogic VHDL design verification obligation
include('Axioms/HWV003-0.ax')
p_Reset(t_{139})   cnf(quest$_1$, negated_conjecture)
¬ gt(fifo_length, rd_level($t_{139} + n_1$))   cnf(quest$_2$, negated_conjecture)
gt(fifo_length, n_0)   cnf(quest$_3$, negated_conjecture)

HWV032-2.p Safelogic VHDL design verification obligation
include('Axioms/HWV004-0.ax')
p__pred_(fwork_DOTfifo_DOTrtl_DOTreset_(t_{206}))   cnf(quest$_1$, negated_conjecture)
p_LES_EQU_(fwork_DOTfifo_DOTrtl_DOTfifo__length_, fwork_DOTfifo_DOTrtl_DOTrd__level_(f_ADD_(t_{206}, n_1)))   cnf(que
¬ p_LES_EQU_(fwork_DOTfifo_DOTrtl_DOTfifo__length_, n_0)   cnf(quest$_3$, negated_conjecture)

HWV033-1.p Safelogic VHDL design verification obligation
include('Axioms/HWV003-0.ax')
¬ gt(rd_level(t_{139}), −fifo_length)   cnf(quest$_1$, negated_conjecture)
gt(rd_level($t_{139} + n_1$), −fifo_length)   cnf(quest$_2$, negated_conjecture)
gt(fifo_length, n_0)   cnf(quest$_3$, negated_conjecture)

HWV033-2.p Safelogic VHDL design verification obligation
include('Axioms/HWV004-0.ax')
p_LES_EQU_(fwork_DOTfifo_DOTrtl_DOTrd__level_(t_{206}), f_SUB_(fwork_DOTfifo_DOTrtl_DOTfifo__length_, n_1))   cnf(ques
¬ p_LES_EQU_(fwork_DOTfifo_DOTrtl_DOTrd__level_(f_ADD_(t_{206}, n_1)), f_SUB_(fwork_DOTfifo_DOTrtl_DOTfifo__length_, n
¬ p_LES_EQU_(fwork_DOTfifo_DOTrtl_DOTfifo__length_, n_0)   cnf(quest$_3$, negated_conjecture)

HWV034-1.p Connections, faults, and gates.
include('Axioms/HWV001-0.ax')

HWV034-2.p Connections, faults, and gates.
include('Axioms/HWV002-0.ax')

HWV035-1.p Half-adder.

include('Axioms/HWV001-0.ax')
include('Axioms/HWV001-1.ax')

HWV035-2.p Half-adder.
include('Axioms/HWV002-0.ax')
include('Axioms/HWV002-1.ax')

HWV036-1.p Full-adder.
include('Axioms/HWV001-0.ax')
include('Axioms/HWV001-1.ax')
include('Axioms/HWV001-2.ax')

HWV036-2.p Full-adder.
include('Axioms/HWV002-0.ax')
include('Axioms/HWV002-1.ax')
include('Axioms/HWV002-2.ax')

HWV037-1.p Axioms from a VHDL design description
include('Axioms/HWV003-0.ax')

HWV038-1.p Axioms from a VHDL design description
include('Axioms/HWV004-0.ax')

HWV052-1.001.001.p Faulty channel 1 1
The problem of sending N bits over a faulty channel that can mutilate any one bit. We can use K extra bits to help us do this. Satisfiable means that it is possible, unsatisfiable means that it is not possible.
$x = o$ or $x = i$ cnf(bit_domain, axiom)
$x^{-1} \neq x$ cnf(bit_inverse, axiom)
$\text{unpack}_1(x_1, \text{pack}_1(x_1)) = x_1$ cnf(unpack$_1$, axiom)
$\text{unpack}_1(x_1^{-1}, \text{pack}_1(x_1)) = x_1$ cnf(unpack1$_{01}$, axiom)
$\text{unpack}_1(x_1, \text{pack}_1(x_1)^{-1}) = x_1$ cnf(unpack1$_{02}$, axiom)

HWV052-1.001.002.p Faulty channel 1 2
The problem of sending N bits over a faulty channel that can mutilate any one bit. We can use K extra bits to help us do this. Satisfiable means that it is possible, unsatisfiable means that it is not possible.
$x = o$ or $x = i$ cnf(bit_domain, axiom)
$x^{-1} \neq x$ cnf(bit_inverse, axiom)
$\text{unpack}_1(x_1, \text{pack}_1(x_1), \text{pack}_2(x_1)) = x_1$ cnf(unpack$_1$, axiom)
$\text{unpack}_1(x_1^{-1}, \text{pack}_1(x_1), \text{pack}_2(x_1)) = x_1$ cnf(unpack1$_{01}$, axiom)
$\text{unpack}_1(x_1, \text{pack}_1(x_1)^{-1}, \text{pack}_2(x_1)) = x_1$ cnf(unpack1$_{02}$, axiom)
$\text{unpack}_1(x_1, \text{pack}_1(x_1), \text{pack}_2(x_1)^{-1}) = x_1$ cnf(unpack1$_{03}$, axiom)

HWV052-1.002.001.p Faulty channel 2 1
The problem of sending N bits over a faulty channel that can mutilate any one bit. We can use K extra bits to help us do this. Satisfiable means that it is possible, unsatisfiable means that it is not possible.
$x = o$ or $x = i$ cnf(bit_domain, axiom)
$x^{-1} \neq x$ cnf(bit_inverse, axiom)
$\text{unpack}_1(x_1, x_2, \text{pack}_1(x_1, x_2)) = x_1$ cnf(unpack$_1$, axiom)
$\text{unpack}_1(x_1^{-1}, x_2, \text{pack}_1(x_1, x_2)) = x_1$ cnf(unpack1$_{01}$, axiom)
$\text{unpack}_1(x_1, x_2^{-1}, \text{pack}_1(x_1, x_2)) = x_1$ cnf(unpack1$_{02}$, axiom)
$\text{unpack}_1(x_1, x_2, \text{pack}_1(x_1, x_2)^{-1}) = x_1$ cnf(unpack1$_{03}$, axiom)
$\text{unpack}_2(x_1, x_2, \text{pack}_1(x_1, x_2)) = x_2$ cnf(unpack$_2$, axiom)
$\text{unpack}_2(x_1^{-1}, x_2, \text{pack}_1(x_1, x_2)) = x_2$ cnf(unpack2$_{04}$, axiom)
$\text{unpack}_2(x_1, x_2^{-1}, \text{pack}_1(x_1, x_2)) = x_2$ cnf(unpack2$_{05}$, axiom)
$\text{unpack}_2(x_1, x_2, \text{pack}_1(x_1, x_2)^{-1}) = x_2$ cnf(unpack2$_{06}$, axiom)

HWV052-1.002.002.p Faulty channel 2 2
The problem of sending N bits over a faulty channel that can mutilate any one bit. We can use K extra bits to help us do this. Satisfiable means that it is possible, unsatisfiable means that it is not possible.
$x = o$ or $x = i$ cnf(bit_domain, axiom)
$x^{-1} \neq x$ cnf(bit_inverse, axiom)
$\text{unpack}_1(x_1, x_2, \text{pack}_1(x_1, x_2), \text{pack}_2(x_1, x_2)) = x_1$ cnf(unpack$_1$, axiom)
$\text{unpack}_1(x_1^{-1}, x_2, \text{pack}_1(x_1, x_2), \text{pack}_2(x_1, x_2)) = x_1$ cnf(unpack1$_{01}$, axiom)
$\text{unpack}_1(x_1, x_2^{-1}, \text{pack}_1(x_1, x_2), \text{pack}_2(x_1, x_2)) = x_1$ cnf(unpack1$_{02}$, axiom)

$\mathrm{unpack}_1(x_1, x_2, \mathrm{pack}_1(x_1, x_2)^{-1}, \mathrm{pack}_2(x_1, x_2)) = x_1$ $\mathrm{cnf}(\mathrm{unpack1}_{03}, \mathrm{axiom})$

$\mathrm{unpack}_1(x_1, x_2, \mathrm{pack}_1(x_1, x_2), \mathrm{pack}_2(x_1, x_2)^{-1}) = x_1$ $\mathrm{cnf}(\mathrm{unpack1}_{04}, \mathrm{axiom})$

$\mathrm{unpack}_2(x_1, x_2, \mathrm{pack}_1(x_1, x_2), \mathrm{pack}_2(x_1, x_2)) = x_2$ $\mathrm{cnf}(\mathrm{unpack}_2, \mathrm{axiom})$

$\mathrm{unpack}_2(x_1^{-1}, x_2, \mathrm{pack}_1(x_1, x_2), \mathrm{pack}_2(x_1, x_2)) = x_2$ $\mathrm{cnf}(\mathrm{unpack2}_{05}, \mathrm{axiom})$

$\mathrm{unpack}_2(x_1, x_2^{-1}, \mathrm{pack}_1(x_1, x_2), \mathrm{pack}_2(x_1, x_2)) = x_2$ $\mathrm{cnf}(\mathrm{unpack2}_{06}, \mathrm{axiom})$

$\mathrm{unpack}_2(x_1, x_2, \mathrm{pack}_1(x_1, x_2)^{-1}, \mathrm{pack}_2(x_1, x_2)) = x_2$ $\mathrm{cnf}(\mathrm{unpack2}_{07}, \mathrm{axiom})$

$\mathrm{unpack}_2(x_1, x_2, \mathrm{pack}_1(x_1, x_2), \mathrm{pack}_2(x_1, x_2)^{-1}) = x_2$ $\mathrm{cnf}(\mathrm{unpack2}_{08}, \mathrm{axiom})$

HWV052-1.002.003.p Faulty channel 2 3

The problem of sending N bits over a faulty channel that can mutilate any one bit. We can use K extra bits to help us do this. Satisfiable means that it is possible, unsatisfiable means that it is not possible.

$x = o$ or $x = i$ $\mathrm{cnf}(\mathrm{bit\_domain}, \mathrm{axiom})$

$x^{-1} \neq x$ $\mathrm{cnf}(\mathrm{bit\_inverse}, \mathrm{axiom})$

$\mathrm{unpack}_1(x_1, x_2, \mathrm{pack}_1(x_1, x_2), \mathrm{pack}_2(x_1, x_2), \mathrm{pack}_3(x_1, x_2)) = x_1$ $\mathrm{cnf}(\mathrm{unpack}_1, \mathrm{axiom})$

$\mathrm{unpack}_1(x_1^{-1}, x_2, \mathrm{pack}_1(x_1, x_2), \mathrm{pack}_2(x_1, x_2), \mathrm{pack}_3(x_1, x_2)) = x_1$ $\mathrm{cnf}(\mathrm{unpack1}_{01}, \mathrm{axiom})$

$\mathrm{unpack}_1(x_1, x_2^{-1}, \mathrm{pack}_1(x_1, x_2), \mathrm{pack}_2(x_1, x_2), \mathrm{pack}_3(x_1, x_2)) = x_1$ $\mathrm{cnf}(\mathrm{unpack1}_{02}, \mathrm{axiom})$

$\mathrm{unpack}_1(x_1, x_2, \mathrm{pack}_1(x_1, x_2)^{-1}, \mathrm{pack}_2(x_1, x_2), \mathrm{pack}_3(x_1, x_2)) = x_1$ $\mathrm{cnf}(\mathrm{unpack1}_{03}, \mathrm{axiom})$

$\mathrm{unpack}_1(x_1, x_2, \mathrm{pack}_1(x_1, x_2), \mathrm{pack}_2(x_1, x_2)^{-1}, \mathrm{pack}_3(x_1, x_2)) = x_1$ $\mathrm{cnf}(\mathrm{unpack1}_{04}, \mathrm{axiom})$

$\mathrm{unpack}_1(x_1, x_2, \mathrm{pack}_1(x_1, x_2), \mathrm{pack}_2(x_1, x_2), \mathrm{pack}_3(x_1, x_2)^{-1}) = x_1$ $\mathrm{cnf}(\mathrm{unpack1}_{05}, \mathrm{axiom})$

$\mathrm{unpack}_2(x_1, x_2, \mathrm{pack}_1(x_1, x_2), \mathrm{pack}_2(x_1, x_2), \mathrm{pack}_3(x_1, x_2)) = x_2$ $\mathrm{cnf}(\mathrm{unpack}_2, \mathrm{axiom})$

$\mathrm{unpack}_2(x_1^{-1}, x_2, \mathrm{pack}_1(x_1, x_2), \mathrm{pack}_2(x_1, x_2), \mathrm{pack}_3(x_1, x_2)) = x_2$ $\mathrm{cnf}(\mathrm{unpack2}_{06}, \mathrm{axiom})$

$\mathrm{unpack}_2(x_1, x_2^{-1}, \mathrm{pack}_1(x_1, x_2), \mathrm{pack}_2(x_1, x_2), \mathrm{pack}_3(x_1, x_2)) = x_2$ $\mathrm{cnf}(\mathrm{unpack2}_{07}, \mathrm{axiom})$

$\mathrm{unpack}_2(x_1, x_2, \mathrm{pack}_1(x_1, x_2)^{-1}, \mathrm{pack}_2(x_1, x_2), \mathrm{pack}_3(x_1, x_2)) = x_2$ $\mathrm{cnf}(\mathrm{unpack2}_{08}, \mathrm{axiom})$

$\mathrm{unpack}_2(x_1, x_2, \mathrm{pack}_1(x_1, x_2), \mathrm{pack}_2(x_1, x_2)^{-1}, \mathrm{pack}_3(x_1, x_2)) = x_2$ $\mathrm{cnf}(\mathrm{unpack2}_{09}, \mathrm{axiom})$

$\mathrm{unpack}_2(x_1, x_2, \mathrm{pack}_1(x_1, x_2), \mathrm{pack}_2(x_1, x_2), \mathrm{pack}_3(x_1, x_2)^{-1}) = x_2$ $\mathrm{cnf}(\mathrm{unpack2}_{10}, \mathrm{axiom})$

HWV052-1.003.003.p Faulty channel 3 3

The problem of sending N bits over a faulty channel that can mutilate any one bit. We can use K extra bits to help us do this. Satisfiable means that it is possible, unsatisfiable means that it is not possible.

$x = o$ or $x = i$ $\mathrm{cnf}(\mathrm{bit\_domain}, \mathrm{axiom})$

$x^{-1} \neq x$ $\mathrm{cnf}(\mathrm{bit\_inverse}, \mathrm{axiom})$

$\mathrm{unpack}_1(x_1, x_2, x_3, \mathrm{pack}_1(x_1, x_2, x_3), \mathrm{pack}_2(x_1, x_2, x_3), \mathrm{pack}_3(x_1, x_2, x_3)) = x_1$ $\mathrm{cnf}(\mathrm{unpack}_1, \mathrm{axiom})$

$\mathrm{unpack}_1(x_1^{-1}, x_2, x_3, \mathrm{pack}_1(x_1, x_2, x_3), \mathrm{pack}_2(x_1, x_2, x_3), \mathrm{pack}_3(x_1, x_2, x_3)) = x_1$ $\mathrm{cnf}(\mathrm{unpack1}_{01}, \mathrm{axiom})$

$\mathrm{unpack}_1(x_1, x_2^{-1}, x_3, \mathrm{pack}_1(x_1, x_2, x_3), \mathrm{pack}_2(x_1, x_2, x_3), \mathrm{pack}_3(x_1, x_2, x_3)) = x_1$ $\mathrm{cnf}(\mathrm{unpack1}_{02}, \mathrm{axiom})$

$\mathrm{unpack}_1(x_1, x_2, x_3^{-1}, \mathrm{pack}_1(x_1, x_2, x_3), \mathrm{pack}_2(x_1, x_2, x_3), \mathrm{pack}_3(x_1, x_2, x_3)) = x_1$ $\mathrm{cnf}(\mathrm{unpack1}_{03}, \mathrm{axiom})$

$\mathrm{unpack}_1(x_1, x_2, x_3, \mathrm{pack}_1(x_1, x_2, x_3)^{-1}, \mathrm{pack}_2(x_1, x_2, x_3), \mathrm{pack}_3(x_1, x_2, x_3)) = x_1$ $\mathrm{cnf}(\mathrm{unpack1}_{04}, \mathrm{axiom})$

$\mathrm{unpack}_1(x_1, x_2, x_3, \mathrm{pack}_1(x_1, x_2, x_3), \mathrm{pack}_2(x_1, x_2, x_3)^{-1}, \mathrm{pack}_3(x_1, x_2, x_3)) = x_1$ $\mathrm{cnf}(\mathrm{unpack1}_{05}, \mathrm{axiom})$

$\mathrm{unpack}_1(x_1, x_2, x_3, \mathrm{pack}_1(x_1, x_2, x_3), \mathrm{pack}_2(x_1, x_2, x_3), \mathrm{pack}_3(x_1, x_2, x_3)^{-1}) = x_1$ $\mathrm{cnf}(\mathrm{unpack1}_{06}, \mathrm{axiom})$

$\mathrm{unpack}_2(x_1, x_2, x_3, \mathrm{pack}_1(x_1, x_2, x_3), \mathrm{pack}_2(x_1, x_2, x_3), \mathrm{pack}_3(x_1, x_2, x_3)) = x_2$ $\mathrm{cnf}(\mathrm{unpack}_2, \mathrm{axiom})$

$\mathrm{unpack}_2(x_1^{-1}, x_2, x_3, \mathrm{pack}_1(x_1, x_2, x_3), \mathrm{pack}_2(x_1, x_2, x_3), \mathrm{pack}_3(x_1, x_2, x_3)) = x_2$ $\mathrm{cnf}(\mathrm{unpack2}_{07}, \mathrm{axiom})$

$\mathrm{unpack}_2(x_1, x_2^{-1}, x_3, \mathrm{pack}_1(x_1, x_2, x_3), \mathrm{pack}_2(x_1, x_2, x_3), \mathrm{pack}_3(x_1, x_2, x_3)) = x_2$ $\mathrm{cnf}(\mathrm{unpack2}_{08}, \mathrm{axiom})$

$\mathrm{unpack}_2(x_1, x_2, x_3^{-1}, \mathrm{pack}_1(x_1, x_2, x_3), \mathrm{pack}_2(x_1, x_2, x_3), \mathrm{pack}_3(x_1, x_2, x_3)) = x_2$ $\mathrm{cnf}(\mathrm{unpack2}_{09}, \mathrm{axiom})$

$\mathrm{unpack}_2(x_1, x_2, x_3, \mathrm{pack}_1(x_1, x_2, x_3)^{-1}, \mathrm{pack}_2(x_1, x_2, x_3), \mathrm{pack}_3(x_1, x_2, x_3)) = x_2$ $\mathrm{cnf}(\mathrm{unpack2}_{10}, \mathrm{axiom})$

$\mathrm{unpack}_2(x_1, x_2, x_3, \mathrm{pack}_1(x_1, x_2, x_3), \mathrm{pack}_2(x_1, x_2, x_3)^{-1}, \mathrm{pack}_3(x_1, x_2, x_3)) = x_2$ $\mathrm{cnf}(\mathrm{unpack2}_{11}, \mathrm{axiom})$

$\mathrm{unpack}_2(x_1, x_2, x_3, \mathrm{pack}_1(x_1, x_2, x_3), \mathrm{pack}_2(x_1, x_2, x_3), \mathrm{pack}_3(x_1, x_2, x_3)^{-1}) = x_2$ $\mathrm{cnf}(\mathrm{unpack2}_{12}, \mathrm{axiom})$

$\mathrm{unpack}_3(x_1, x_2, x_3, \mathrm{pack}_1(x_1, x_2, x_3), \mathrm{pack}_2(x_1, x_2, x_3), \mathrm{pack}_3(x_1, x_2, x_3)) = x_3$ $\mathrm{cnf}(\mathrm{unpack}_3, \mathrm{axiom})$

$\mathrm{unpack}_3(x_1^{-1}, x_2, x_3, \mathrm{pack}_1(x_1, x_2, x_3), \mathrm{pack}_2(x_1, x_2, x_3), \mathrm{pack}_3(x_1, x_2, x_3)) = x_3$ $\mathrm{cnf}(\mathrm{unpack3}_{13}, \mathrm{axiom})$

$\mathrm{unpack}_3(x_1, x_2^{-1}, x_3, \mathrm{pack}_1(x_1, x_2, x_3), \mathrm{pack}_2(x_1, x_2, x_3), \mathrm{pack}_3(x_1, x_2, x_3)) = x_3$ $\mathrm{cnf}(\mathrm{unpack3}_{14}, \mathrm{axiom})$

$\mathrm{unpack}_3(x_1, x_2, x_3^{-1}, \mathrm{pack}_1(x_1, x_2, x_3), \mathrm{pack}_2(x_1, x_2, x_3), \mathrm{pack}_3(x_1, x_2, x_3)) = x_3$ $\mathrm{cnf}(\mathrm{unpack3}_{15}, \mathrm{axiom})$

$\mathrm{unpack}_3(x_1, x_2, x_3, \mathrm{pack}_1(x_1, x_2, x_3)^{-1}, \mathrm{pack}_2(x_1, x_2, x_3), \mathrm{pack}_3(x_1, x_2, x_3)) = x_3$ $\mathrm{cnf}(\mathrm{unpack3}_{16}, \mathrm{axiom})$

$\mathrm{unpack}_3(x_1, x_2, x_3, \mathrm{pack}_1(x_1, x_2, x_3), \mathrm{pack}_2(x_1, x_2, x_3)^{-1}, \mathrm{pack}_3(x_1, x_2, x_3)) = x_3$ $\mathrm{cnf}(\mathrm{unpack3}_{17}, \mathrm{axiom})$

$\mathrm{unpack}_3(x_1, x_2, x_3, \mathrm{pack}_1(x_1, x_2, x_3), \mathrm{pack}_2(x_1, x_2, x_3), \mathrm{pack}_3(x_1, x_2, x_3)^{-1}) = x_3$ $\mathrm{cnf}(\mathrm{unpack3}_{18}, \mathrm{axiom})$