MACHINE
BIT_VECTOR_DEFINITION

SEES
BIT_DEFINITION

CONSTANTS
BIT_VECTOR,
bv_size, *bv_zero*, *bv_one*,
bv_not, *bv_and*, *bv_or*, *bv_xor*,
bv_set, *bv_clear*

PROPERTIES
$BIT\_VECTOR = \mathbf{seq1}\ (BIT) \wedge$

$bv\_size \in BIT\_VECTOR \rightarrow \mathcal{N}_1 \wedge$
$bv\_size = \lambda\ (bv).(bv \in BIT\_VECTOR \mid \mathbf{size}(bv)) \wedge$

$bv\_zero \in \mathcal{N}_1 \rightarrow BIT\_VECTOR \wedge$
$bv\_zero = \lambda\ (sz)\ .\ (sz \in \mathcal{N}_1 \mid (0\ ..\ (sz\text{-}1)) \times \{0\}) \wedge$

$bv\_one \in \mathcal{N}_1 \rightarrow BIT\_VECTOR \wedge$
$bv\_one = \lambda\ (sz)\ .\ (sz \in \mathcal{N}_1 \mid (0\ ..\ (sz\text{-}1)) \times \{1\}) \wedge$

$bv\_not \in BIT\_VECTOR \rightarrow BIT\_VECTOR \wedge$
$bv\_not = \lambda\ (v1).(v1 \in BIT\_VECTOR \mid \lambda\ (idx).(idx \in 0\ ..\ (\mathbf{size}(v1)\text{-}1) \mid bit\_not(v1(idx)))) \wedge$

$bv\_and \in BIT\_VECTOR \times BIT\_VECTOR \rightarrow BIT\_VECTOR \wedge$
$bv\_and = \lambda\ (v1,v2).(v1 \in BIT\_VECTOR \wedge v2 \in BIT\_VECTOR \wedge \mathbf{size}(v1) = \mathbf{size}(v2) \mid$
$\quad \lambda\ (idx).(idx \in 0\ ..\ (\mathbf{size}(v1)\text{-}1) \mid bit\_and(v1(idx),v2(idx)))) \wedge$

$bv\_or \in BIT\_VECTOR \times BIT\_VECTOR \rightarrow BIT\_VECTOR \wedge$
$bv\_or = \lambda\ (v1,v2).(v1 \in BIT\_VECTOR \wedge v2 \in BIT\_VECTOR \wedge \mathbf{size}(v1) = \mathbf{size}(v2) \mid$
$\quad \lambda\ (idx).(idx \in 0\ ..\ (\mathbf{size}(v1)\text{-}1) \mid bit\_or(v1(idx),v2(idx)))) \wedge$

$bv\_xor \in BIT\_VECTOR \times BIT\_VECTOR \rightarrow BIT\_VECTOR \wedge$
$bv\_xor = \lambda\ (v1,v2).(v1 \in BIT\_VECTOR \wedge v2 \in BIT\_VECTOR \wedge \mathbf{size}(v1) = \mathbf{size}(v2) \mid$
$\quad \lambda\ (idx).(idx \in 0\ ..\ (\mathbf{size}(v1)\text{-}1) \mid bit\_xor(v1(idx),v2(idx)))) \wedge$

$bv\_set \in BIT\_VECTOR \times \mathcal{N} \rightarrow BIT\_VECTOR \wedge$
$bv\_set = \lambda\ (v1,idx).(v1 \in BIT\_VECTOR \wedge idx \in \mathcal{N} \wedge idx < \mathbf{size}(v1) \mid v1 \leftarrow \{\ idx \mapsto 1\ \}) \wedge$
$bv\_clear \in BIT\_VECTOR \times \mathcal{N} \rightarrow BIT\_VECTOR \wedge$
$bv\_clear = \lambda\ (v1,idx).(v1 \in BIT\_VECTOR \wedge idx \in \mathcal{N} \wedge idx < \mathbf{size}(v1) \mid v1 \leftarrow \{\ idx \mapsto 0\ \})$

ASSERTIONS
$\forall\ bv.\ (bv \in BIT\_VECTOR \Rightarrow bv\_size(bv\_not(bv)) = bv\_size(bv));$
$\forall\ bv.\ (bv \in BIT\_VECTOR \Rightarrow (bv\_not(bv\_not(bv)) = bv));$

$\forall\ (v1,v2).\ (v1 \in BIT\_VECTOR \wedge v2 \in BIT\_VECTOR \wedge bv\_size(v1) = bv\_size(v2) \Rightarrow$

$bv\_size(bv\_and(v1,\ v2)) = bv\_size(v1))$;

$\forall\ (v1,v2).\ (v1 \in BIT\_VECTOR \wedge v2 \in BIT\_VECTOR \wedge bv\_size(v1) = bv\_size(v2) \Rightarrow$
$bv\_size(bv\_and(v1,\ v2)) = bv\_size(v2))$;

$\forall\ (v1,v2).\ (v1 \in BIT\_VECTOR \wedge v2 \in BIT\_VECTOR \wedge bv\_size(v1) = bv\_size(v2) \Rightarrow$
$(bv\_and(v1,\ v2) = bv\_and(v2,\ v1)))$;

$\forall\ (v1,v2,v3).\ (v1 \in BIT\_VECTOR \wedge v2 \in BIT\_VECTOR \wedge v3 \in BIT\_VECTOR \wedge bv\_size(v1) = bv\_size(v2) \wedge bv\_size(v1) = bv\_size(v3) \Rightarrow$
$(bv\_and(v1,\ bv\_and(v2,v3)) = bv\_and(bv\_and(v1,\ v2),\ v3)))$;

$\forall\ bv.\ (bv \in BIT\_VECTOR \Rightarrow (bv\_and(bv,\ bv\_zero(bv\_size(bv))) = bv\_zero(bv\_size(bv))))$;

$\forall\ bv.\ (bv \in BIT\_VECTOR \Rightarrow (bv\_and(bv,\ bv\_one(bv\_size(bv))) = bv))$;

$\forall\ (v1,v2).\ (v1 \in BIT\_VECTOR \wedge v2 \in BIT\_VECTOR \wedge bv\_size(v1) = bv\_size(v2) \Rightarrow$
$bv\_size(bv\_or(v1,\ v2)) = bv\_size(v1))$;

$\forall\ (v1,v2).\ (v1 \in BIT\_VECTOR \wedge v2 \in BIT\_VECTOR \wedge bv\_size(v1) = bv\_size(v2) \Rightarrow$
$bv\_size(bv\_or(v1,\ v2)) = bv\_size(v2))$;

$\forall\ (v1,v2).\ (v1 \in BIT\_VECTOR \wedge v2 \in BIT\_VECTOR \wedge bv\_size(v1) = bv\_size(v2) \Rightarrow$
$(bv\_or(v1,\ v2) = bv\_or(v2,\ v1)))$;

$\forall\ (v1,v2,v3).\ (v1 \in BIT\_VECTOR \wedge v2 \in BIT\_VECTOR \wedge v3 \in BIT\_VECTOR \wedge bv\_size(v1) = bv\_size(v2) \wedge bv\_size(v1) = bv\_size(v3) \Rightarrow$
$(bv\_or(v1,\ bv\_or(v2,v3)) = bv\_or(bv\_or(v1,\ v2),\ v3)))$;

$\forall\ bv.\ (bv \in BIT\_VECTOR \Rightarrow (bv\_or(bv,\ bv\_one(bv\_size(bv))) = bv\_one(bv\_size(bv))))$;

$\forall\ bv.\ (bv \in BIT\_VECTOR \Rightarrow (bv\_or(bv,\ bv\_zero(bv\_size(bv))) = bv))$;

$\forall\ (v1,v2).\ (v1 \in BIT\_VECTOR \wedge v2 \in BIT\_VECTOR \wedge bv\_size(v1) = bv\_size(v2) \Rightarrow$
$bv\_size(bv\_xor(v1,\ v2)) = bv\_size(v1))$;

$\forall\ (v1,v2).\ (v1 \in BIT\_VECTOR \wedge v2 \in BIT\_VECTOR \wedge bv\_size(v1) = bv\_size(v2) \Rightarrow$
$bv\_size(bv\_xor(v1,\ v2)) = bv\_size(v2))$;

$\forall\ (v1,v2).\ (v1 \in BIT\_VECTOR \wedge v2 \in BIT\_VECTOR \wedge bv\_size(v1) = bv\_size(v2) \Rightarrow$
$(bv\_xor(v1,\ v2) = bv\_xor(v2,\ v1)))$;

$\forall\ (bv).\ (bv \in BIT\_VECTOR \Rightarrow bv\_xor(bv,bv) = bv\_zero(bv\_size(bv)))$

END