MACHINE
ALU

SEES
TYPES,
BIT_DEFINITION,
BIT_VECTOR_DEFINITION,
BIT_VECTOR_ARITHMETICS,
BYTE_DEFINITION

CONSTANTS
add, *substract*,
and, *ior*, *xor*,
bitclear, *bitset*, *bitget*,
complement, *swap*, *rotateleft*, *rotateright*

PROPERTIES

$add : (UCHAR \times UCHAR) \rightarrow (UCHAR \times \mathbf{BOOL} \times \mathbf{BOOL} \times \mathbf{BOOL}) \wedge$
$\forall\ (w1, w2, sum).$
$(w1 \in UCHAR \wedge w2 \in UCHAR \wedge sum \in \mathcal{N} \wedge sum = w1{+}w2 \Rightarrow$
$((sum \leq 255 \Rightarrow add(w1, w2) = (sum, \mathbf{bool}(sum{=}0), \mathbf{bool}(w1\ /\ 16 + w2\ /\ 16 > 15), \mathbf{FALSE})) \wedge$
$(256 \leq sum \Rightarrow add(w1, w2) = (sum\text{-}256, \mathbf{bool}(sum{=}256), \mathbf{bool}(w1\ /\ 16 + w2\ /\ 16 > 15), \mathbf{TRUE})))) \wedge$

$substract : (UCHAR \times UCHAR) \rightarrow (UCHAR \times \mathbf{BOOL} \times \mathbf{BOOL}) \wedge$
$\forall\ (w1, w2, diff).$
$(w1 \in UCHAR \wedge w2 \in UCHAR \wedge diff \in \mathcal{Z} \wedge diff = w1\text{-}w2 \Rightarrow$
$((diff < 0 \Rightarrow substract(w1, w2) = (diff{+}256, \mathbf{FALSE}, \mathbf{TRUE})) \wedge$
$(diff \geq 0 \Rightarrow substract(w1, w2) = (diff, \mathbf{bool}(diff{=}0), \mathbf{FALSE})))) \wedge$
$and : (BYTE \times BYTE) \rightarrow (BYTE \times \mathbf{BOOL}) \wedge$
$\forall\ (w1, w2, ww).$
$(w1 \in BYTE \wedge w2 \in BYTE \wedge ww \in BYTE \wedge ww = bv\_and(w1, w2) \Rightarrow$
$and(w1, w2) = (ww, \mathbf{bool}(bv\_to\_nat(ww) = 0))) \wedge$
$ior : (BYTE \times BYTE) \rightarrow (BYTE \times \mathbf{BOOL}) \wedge$
$\forall\ (w1, w2, ww).$
$(w1 \in BYTE \wedge w2 \in BYTE \wedge ww \in BYTE \wedge ww = bv\_or(w1, w2) \Rightarrow$
$ior(w1, w2) = (ww, \mathbf{bool}(bv\_to\_nat(ww) = 0))) \wedge$
$xor : (BYTE \times BYTE) \rightarrow (BYTE \times \mathbf{BOOL}) \wedge$
$\forall\ (w1, w2, ww).$
$(w1 \in BYTE \wedge w2 \in BYTE \wedge ww \in BYTE \Rightarrow$
$(ww = bv\_xor(w1, w2) \Rightarrow$
$xor(w1, w2) = (ww, \mathbf{bool}(bv\_to\_nat(ww) = 0)))) \wedge$
$bitget : (BYTE \times BYTE\_INDEX) \rightarrow BIT \wedge$
$\forall\ (ww, ii).(ww \in BYTE \wedge ii \in BYTE\_INDEX \Rightarrow bitget(ww,ii) = ww(ii)) \wedge$
$bitset : (BYTE \times BYTE\_INDEX) \rightarrow BYTE \wedge$
$\forall\ (ww, ii).(ww \in BYTE \wedge ii \in BYTE\_INDEX \Rightarrow bitset(ww,ii) = bv\_set(ww, ii)) \wedge$

$bitclear : (BYTE \times BYTE\_INDEX) \rightarrow BYTE \wedge$

$\forall\ (ww,\ ii,\ bb).(ww \in BYTE \wedge ii \in BYTE\_INDEX \wedge bb \in BIT \Rightarrow bitclear(ww,ii) = bv\_clear(ww,\ ii)) \wedge$

$complement \in BYTE \rightarrow BYTE \wedge$

$\forall\ (ww).(ww \in BYTE \Rightarrow complement(ww) = bv\_not(ww)) \wedge$

$swap \in BYTE \rightarrow BYTE \wedge$

$\forall\ (ww).(ww \in BYTE \Rightarrow$

$(swap(ww) = \{0 \mapsto ww(4),\ 1 \mapsto ww(5),\ 2 \mapsto ww(6),\ 3 \mapsto ww(7),\ 4 \mapsto ww(0),\ 5 \mapsto ww(1),\ 6 \mapsto ww(2),\ 7 \mapsto ww(3)\})) \wedge$

$rotateleft \in BYTE \rightarrow BYTE \times \mathbf{BOOL} \wedge$

$\forall\ (ww).(ww \in BYTE \Rightarrow$

$(rotateleft(ww) = (\{0 \mapsto ww(7),\ 1 \mapsto ww(0),\ 2 \mapsto ww(1),\ 3 \mapsto ww(2),\ 4 \mapsto ww(3),\ 5 \mapsto ww(4),\ 6 \mapsto ww(5),\ 7 \mapsto ww(6)\},$

$\mathbf{bool}(ww(7){=}1)))) \wedge$

$rotateright \in BYTE \rightarrow BYTE \times \mathbf{BOOL} \quad \wedge$

$\forall\ (ww).(ww \in BYTE \quad \Rightarrow$

$(rotateright(ww) = (\{0 \mapsto ww(1),\ 1 \mapsto ww(2),\ 2 \mapsto ww(3),\ 3 \mapsto ww(4),\ 4 \mapsto ww(5),\ 5 \mapsto ww(6),\ 6 \mapsto ww(7),\ 7 \mapsto ww(0)\},$

$\mathbf{bool}(ww(0){=}1))))$

ASSERTIONS

$\mathbf{dom}(add) = UCHAR \times UCHAR;$

$\mathbf{ran}(add) \subseteq UCHAR \times \mathbf{BOOL} \times \mathbf{BOOL} \times \mathbf{BOOL};$

$\mathbf{dom}(substract) = UCHAR \times UCHAR;$

$\mathbf{ran}(substract) \subseteq UCHAR \times \mathbf{BOOL} \times \mathbf{BOOL};$

$\mathbf{dom}(and) = BYTE \times BYTE;$

$\mathbf{ran}(and) \subseteq BYTE \times \mathbf{BOOL};$

$\mathbf{dom}(ior) = BYTE \times BYTE;$

$\mathbf{ran}(ior) \subseteq BYTE \times \mathbf{BOOL};$

$\mathbf{dom}(xor) = BYTE \times BYTE;$

$\mathbf{ran}(xor) \subseteq BYTE \times \mathbf{BOOL};$

$\mathbf{dom}(bitclear) = BYTE \times BYTE\_INDEX;$

$\mathbf{ran}(bitclear) \subseteq BYTE;$

$\mathbf{dom}(bitset) = BYTE \times BYTE\_INDEX;$

$\mathbf{ran}(bitset) \subseteq BYTE;$

$\mathbf{dom}(bitget) = BYTE \times BYTE\_INDEX;$

$\mathbf{ran}(bitget) \subseteq BIT;$

$\mathbf{dom}(complement) = BYTE;$

$\mathbf{ran}(complement) \subseteq BYTE;$

$\mathbf{dom}(swap) = BYTE;$

$\mathbf{ran}(swap) \subseteq BYTE;$

$\mathbf{ran}(rotateleft) \subseteq BYTE \times \mathbf{BOOL};$

$\mathbf{dom}(rotateleft) = BYTE;$

$\mathbf{dom}(rotateright) = BYTE;$

$\mathbf{ran}(rotateright) \subseteq BYTE \times \mathbf{BOOL}$

END