MODULE *GreyCodeCounter*

EXTENDS *Sequences*, *Integers*, *TLC*

Grey Code is a way of encoding binary information such that no two bits flip in one step. This is used in Flash Memory to minimize bit flips during self-discharge

```
--algorithm GreyCode{

variables
    flashCell ∈ {⟨0, 0⟩, ⟨0, 1⟩, ⟨1, 1⟩, ⟨1, 0⟩} ;
    actions = [before ↦ ⟨⟩, after ↦ ⟨⟩] ;
{

while ( TRUE ) {
    actions.before := flashCell ;
    if ( flashCell = ⟨0, 0⟩ ) {

        flashCell := ⟨0, 1⟩ ;
        actions.after := ⟨0, 1⟩ ;

    } else if ( flashCell = ⟨0, 1⟩ ) {
        flashCell := ⟨1, 1⟩ ;
        actions.after := ⟨1, 1⟩ ;

    } else if ( flashCell = ⟨1, 0⟩ ) {

        flashCell := ⟨0, 0⟩ ;
        actions.after := ⟨0, 0⟩ ;

    } else {

        flashCell := ⟨1, 0⟩ ;
        actions.after := ⟨1, 0⟩ ;

    }

}
}
}
```

end algorithm;

BEGIN TRANSLATION ($chksum(pcal) =$ "$f604e792$" $\land\ chksum(tla) =$ "$ce051695$")

VARIABLES *flashCell*, *actions*, *pc*

$vars \triangleq \langle flashCell, actions, pc\rangle$

$Init \triangleq$ Global variables

$\land\ flashCell \in \{\langle 0, 0\rangle, \langle 0, 1\rangle, \langle 1, 1\rangle, \langle 1, 0\rangle\}$

$\land\ actions = [before \mapsto \langle\rangle, after \mapsto \langle\rangle]$

$\land\ pc =$ "Lbl_1"

$$
\begin{array}{l}
Lbl\_1 \triangleq \\
\quad \land\ pc = \text{“Lbl\_1”} \\
\quad \land\ actions' = [actions \text{ EXCEPT } !.before = flashCell] \\
\quad \land \text{ IF } flashCell = \langle 0,\ 0 \rangle \\
\qquad \text{THEN } \land\ flashCell' = \langle 0,\ 1 \rangle \\
\qquad\qquad\ \land\ pc' = \text{“Lbl\_2”} \\
\qquad \text{ELSE } \land \text{ IF } flashCell = \langle 0,\ 1 \rangle \\
\qquad\qquad\qquad \text{THEN } \land\ flashCell' = \langle 1,\ 1 \rangle \\
\qquad\qquad\qquad\qquad\ \land\ pc' = \text{“Lbl\_3”} \\
\qquad\qquad\qquad \text{ELSE } \land \text{ IF } flashCell = \langle 1,\ 0 \rangle \\
\qquad\qquad\qquad\qquad\qquad \text{THEN } \land\ flashCell' = \langle 0,\ 0 \rangle \\
\qquad\qquad\qquad\qquad\qquad\qquad\ \land\ pc' = \text{“Lbl\_4”} \\
\qquad\qquad\qquad\qquad\qquad \text{ELSE } \land\ flashCell' = \langle 1,\ 0 \rangle \\
\qquad\qquad\qquad\qquad\qquad\qquad\ \land\ pc' = \text{“Lbl\_5”}
\end{array}
$$

$$
\begin{array}{l}
Lbl\_2 \triangleq \\
\quad \land\ pc = \text{“Lbl\_2”} \\
\quad \land\ actions' = [actions \text{ EXCEPT } !.after = \langle 0,\ 1 \rangle] \\
\quad \land\ pc' = \text{“Lbl\_1”} \\
\quad \land \text{ UNCHANGED } flashCell
\end{array}
$$

$$
\begin{array}{l}
Lbl\_3 \triangleq \\
\quad \land\ pc = \text{“Lbl\_3”} \\
\quad \land\ actions' = [actions \text{ EXCEPT } !.after = \langle 1,\ 1 \rangle] \\
\quad \land\ pc' = \text{“Lbl\_1”} \\
\quad \land \text{ UNCHANGED } flashCell
\end{array}
$$

$$
\begin{array}{l}
Lbl\_4 \triangleq \\
\quad \land\ pc = \text{“Lbl\_4”} \\
\quad \land\ actions' = [actions \text{ EXCEPT } !.after = \langle 0,\ 0 \rangle] \\
\quad \land\ pc' = \text{“Lbl\_1”} \\
\quad \land \text{ UNCHANGED } flashCell
\end{array}
$$

$$
\begin{array}{l}
Lbl\_5 \triangleq \\
\quad \land\ pc = \text{“Lbl\_5”} \\
\quad \land\ actions' = [actions \text{ EXCEPT } !.after = \langle 1,\ 0 \rangle] \\
\quad \land\ pc' = \text{“Lbl\_1”} \\
\quad \land \text{ UNCHANGED } flashCell
\end{array}
$$

$$Next \triangleq Lbl\_1 \lor Lbl\_2 \lor Lbl\_3 \lor Lbl\_4 \lor Lbl\_5$$

$$Spec \triangleq Init \land \Box[Next]_{vars}$$

END TRANSLATION

$$TypeOk \triangleq flashCell[1] \in \{0,\ 1\} \land flashCell[2] \in \{0,\ 1\}$$ They are 1 indexed!

Only 1 bit can change between two states. $\langle 0,\ 0 \rangle \to \langle 1,\ 1 \rangle$ is an illegal transition. Hence, “distance” between any two states is $|\{0,\ 1\}|$

$$
\begin{array}{l}
OneBitAtATime \triangleq \text{ IF } actions.before \neq \langle\rangle \land actions.after \neq \langle\rangle \\
\quad \text{THEN} \\
\qquad (actions.after[1] + actions.after[2]) - (actions.before[1] + actions.before[2]) \ \in \{-1,\ 0,\ 1\}
\end{array}
$$

```
    ELSE   TRUE

```

\ * Modification History
\ * Last modified *Tue Mar* 02 18:33:42 *MST* 2021 by *jeremy*
\ * Created *Tue Mar* 02 16:05:58 *MST* 2021 by *jeremy*