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

end algorithm;

BEGIN TRANSLATION (chksum(pcal) = "58c9e3f0" ∧ chksum(tla) = "8860fb5f")
VARIABLES flashCell, actions, pc

vars ≜ ⟨flashCell, actions, pc⟩

Init ≜ Global variables
  ∧ flashCell ∈ {⟨0, 0⟩, ⟨0, 1⟩, ⟨1, 1⟩, ⟨1, 0⟩}
  ∧ actions = [before ↦ ⟨⟩, after ↦ ⟨⟩]
  ∧ pc = "Lbl_1"

Lbl_1 ≜ ∧ pc = "Lbl_1"
  ∧ IF flashCell = ⟨0, 0⟩
    THEN ∧ actions' = [actions EXCEPT !.before = ⟨0, 0⟩]
      ∧ flashCell' = ⟨1, 1⟩
      ∧ pc' = "Lbl_2"
    ELSE ∧ actions' = [actions EXCEPT !.before = ⟨0, 0⟩]
      ∧ pc' = "Lbl_3"
      ∧ UNCHANGED flashCell

Lbl_3 ≜ ∧ pc = "Lbl_3"
  ∧ flashCell' = ⟨1, 1⟩

```

```

       $\wedge actions' = [actions \text{ EXCEPT } !.after = \langle 1, 1 \rangle]$ 
       $\wedge pc' = \text{"Lbl\_1"}$ 

Lbl_2  $\triangleq$   $\wedge pc = \text{"Lbl\_2"}$ 
       $\wedge actions' = [actions \text{ EXCEPT } !.after = \langle 1, 1 \rangle]$ 
       $\wedge pc' = \text{"Lbl\_3"}$ 
       $\wedge \text{UNCHANGED } flashCell$ 

Next  $\triangleq$  Lbl_1  $\vee$  Lbl_3  $\vee$  Lbl_2

Spec  $\triangleq$  Init  $\wedge \square[Next]_{vars}$ 

END TRANSLATION

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

Only 1 bit can change between two states.  $\langle 0, 0 \rangle \rightarrow \langle 1, 1 \rangle$  is an illegal transition
OneBitAtATime  $\triangleq$  IF actions.before  $\neq \langle \rangle \wedge actions.after \neq \langle \rangle$ 
  THEN
    (actions.after[1] + actions.after[2]) - (actions.before[1] + actions.after[2])  $\in \{1\}$ 
  ELSE TRUE

```

```

\ * Modification History
\ * Last modified Tue Mar 02 17:00:40 MST 2021 by jeremy
\ * Created Tue Mar 02 16:05:58 MST 2021 by jeremy

```