Set a byte to non-zero

by White Flame

Consider a flag in memory that is initialized to zero, but should become non-zero based on some check. The byte will later be polled to invoke some behavior and reset to zero. Sometimes small/fast programs need to get clever with this very simple operation depending on how the registers are constrained.

In this page, the “MNZ” operation means “Make Non-Zero”.

Using Register Values

If a register is known to contain a non-zero value (or is free to immediately set its value), then a basic STA/STX/STY will MNZ. This is the only way to perform this operation without affecting the N or Z processor flags.

```
  STA flag
```

However, sometimes our register values are unknown and must be preserved.

Rolling a Carry Bit

ROL/ROR'ing in a set (or known set) carry bit will guarantee a byte becomes non-zero without affecting the registers. This does destroy the carry bit, unless it is guaranteed that MNZ will not be performed on a flag more than 7 times before checking & resetting to zero.

```
SEC
ROL flag
```

However, sometimes the carry state is unknown and must be preserved.

Using INC/DEC

Using INC or DEC to pull a value away from zero allows a flag to be MNZ'd up to 255 times, while preserving all of A, X, Y, and the Carry flag.

```
 INC flag
```

However, sometimes we do not know how often the MNZ operation will be done between resets.

This final means ALWAYS guarantees a flag byte becomes non-zero without affecting A, X, Y, or C, no matter how often it's run between resets:

```
: INC flag
```

```
  BEQ :-
```

The branch will be taken incredibly rarely.