UNIVERSITÀ
DI TRENTO

Department of Engineering and Information Science

Course of Embedded Systems

NON VOLATILE EMULATOR FOR FPGA

Caronti Luca 192298
Ruffini Simone 188101

Accademic year 2019/2020

Contents

Chapter 1

Abstract

The goal of this project was to create a Non Volatile Emulator (NVE), written in VHDL for FPGAs. The theoretical concept was taken from paper *Emulating Intermittent Non-Volatile Computing Systems Using Off-the-shlef FPGAs* of Davide Brunelli and Kasım Sinan Yıldırım.
There are 4 main modules:

- **Power Approximator:** It's basically a counter that counts how many clock cycles each power state is enable.
- **Instant Power Calculator:** Transforms Power Approximator counters into a Power value.
- **Intermittency Emulator:** Emulates the intermittency due to lack of energy.
- **NV Register Emulator:**

These 4 modules are needed to emulate the behavior of the entity under test. In this case it's a simple **adder** that reads a value from BRAM, adds one to this value and saves it's again in the BRAM, and so on.

Chapter 2

Project Overview

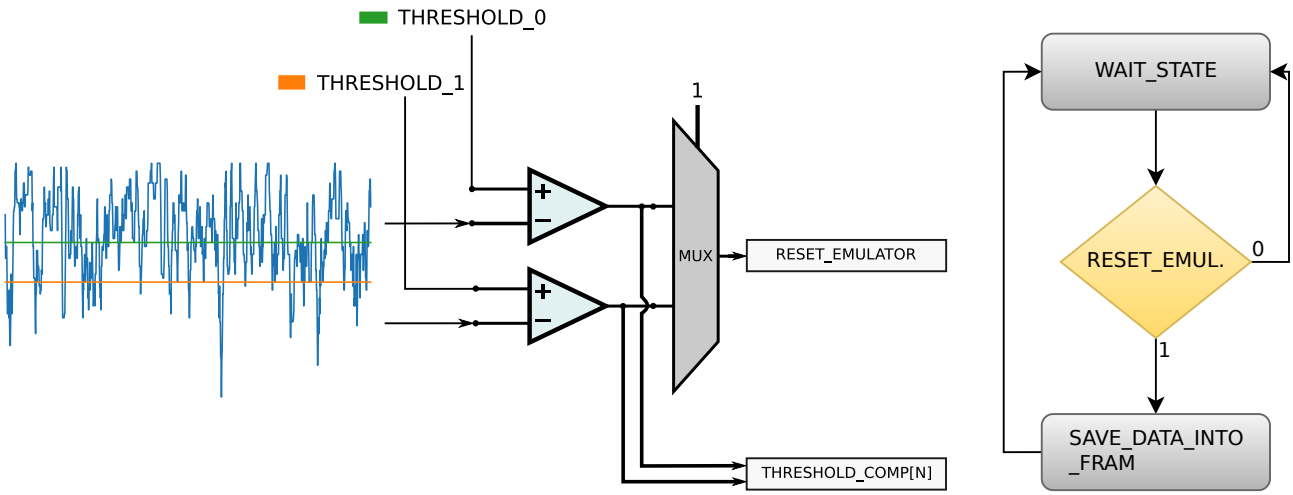

Figure 2.1: General concept

Chapter 3

Architecture

3.1 Power Approximator

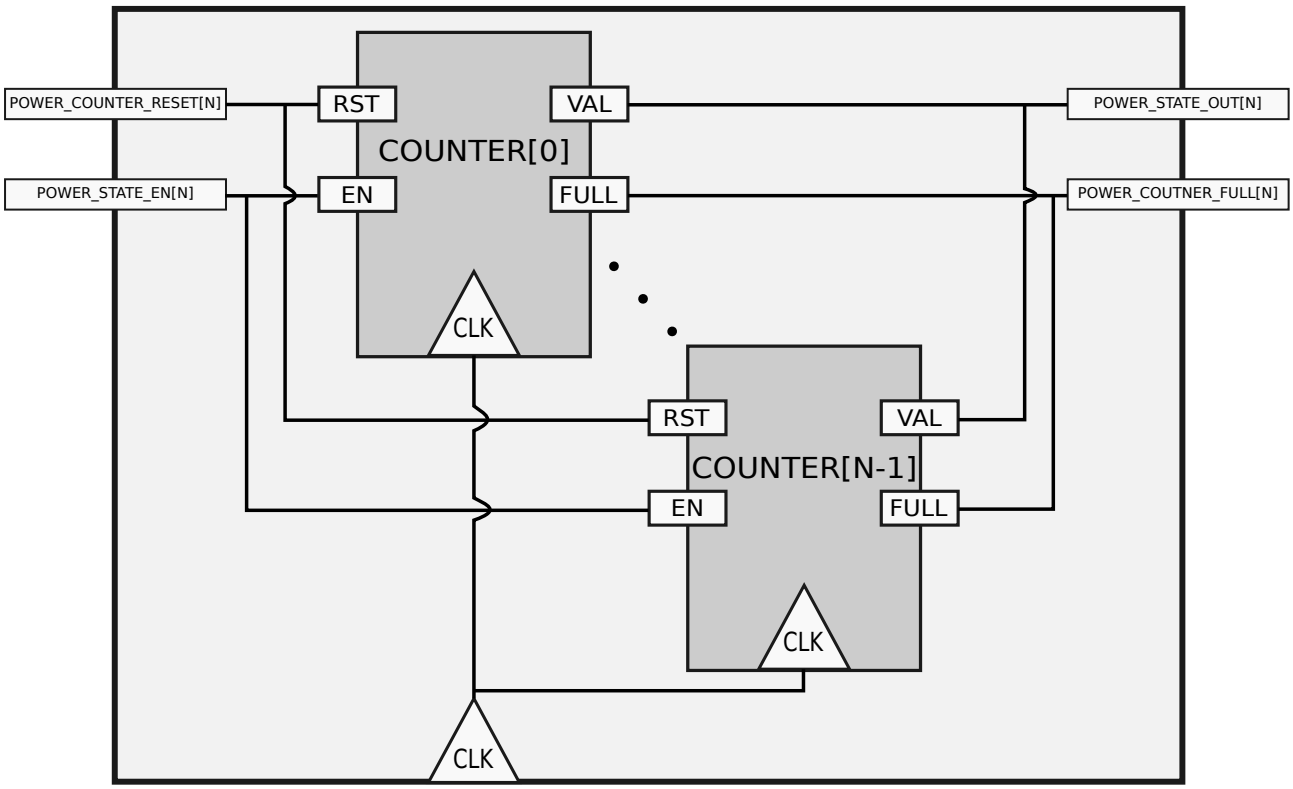

Figure 3.1: Power Approximator Architecture

Power Approximator (PA) (Figure 3.1) is used to calculate how many clock cycles each power state is enable. There are N counter, one for each state, that are instantiate automatically as shown below.

```
GEN_COUNTERS : for i in 0 to NUM_PWR_STATE - 1 generate
    COUTER : counter
        generic map(
            MAX          => 2**COUNTER_MAX_NUM_BIT-1,
            INIT_VALUE   => 0,
            INCREASE_BY => 1
        )
        port map(
            clk          => sys_clk,
            INIT         => power_counter_reset(i),
            CE           => power_state_en(i),
```

```
            TC          => power_counter_full(i),
            value       => power_state_out(i)
        );
end generate;
```

It's possible to see that counter generated are equal to constant **NUM_PWR_STATE** that are located in file called **global_settings.vhd**. The max value of counter is set with constant **COUNTER_MAX_NUM_BIT**, always located in the previous cited file, and it's equal to:

$$\text{Max counter val} = 2^{\text{COUNTER\_MAX\_NUM\_BIT}}$$

3.2 Instant Power Calculator

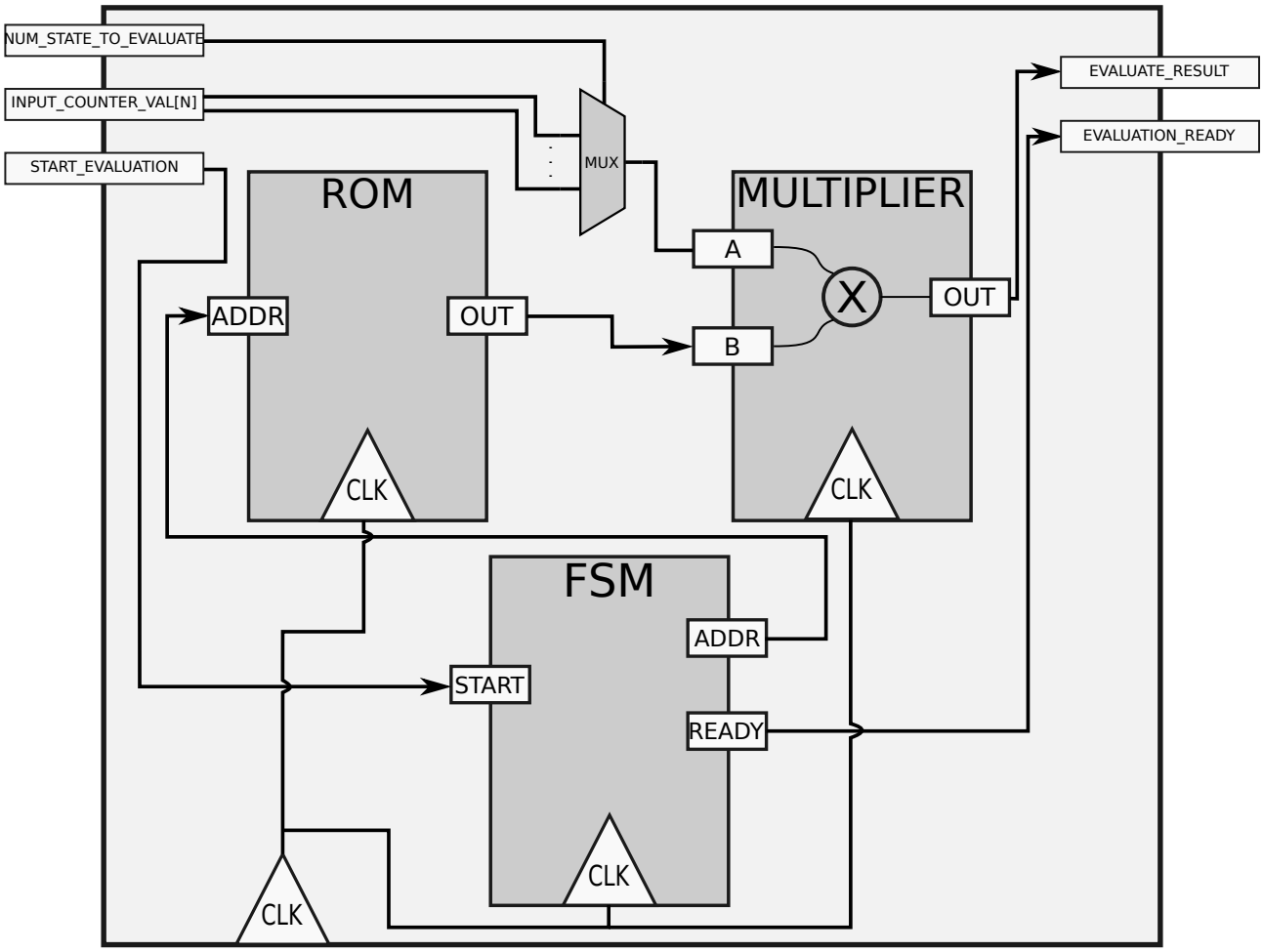

Figure 3.2: Instant power calculator architecture

Instant Power Calculator (IPC) (Figure 3.2) is used to convert the counter value of PA into a power value. It is performed multiplying the counter value with the corresponding power consumption for clock cycle (PCCC).

$$\text{PWR}[W] = \text{COUNTER\_VAL}[CLK] \cdot \Delta\text{PWR\_CONSUMPTION} \left[\frac{W}{CKL}\right]$$

All PCCC values are stored in a ROM, with the address corresponding to the number of the state. All operations, like reading values from ROM and coordinate the multiplication, are managed by a finite state machine. The latter also sets the *EVALUATION_READY* signal for one clock cycle when the evaluation is performed.

3.3 Intermittency Emulator

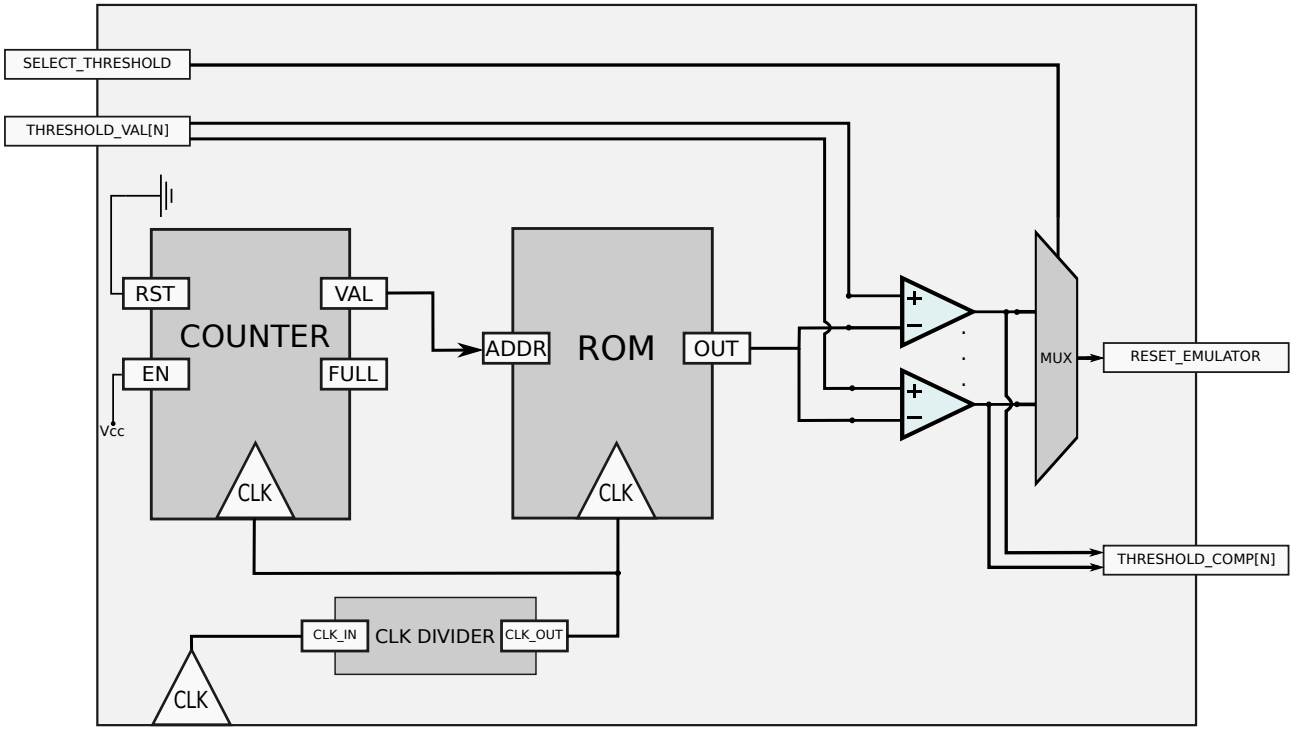

Figure 3.3: Intermittency Emulator Architecture

Itermittency Emulator (IE) (figure 3.3) is used to emulate a real case of lack of energy, in fact there is a ROM that contains a feigning voltage trace. User can set a variable number of threshold to compare with the voltage trace. One of those can be used as reset for the entity under test. The voltage trace values change each clock cycle (divided by a prescaler) thanks to a counter that is always-on.

3.4 Non Volatile Register Emulator

3.5 Adder (entity under test)

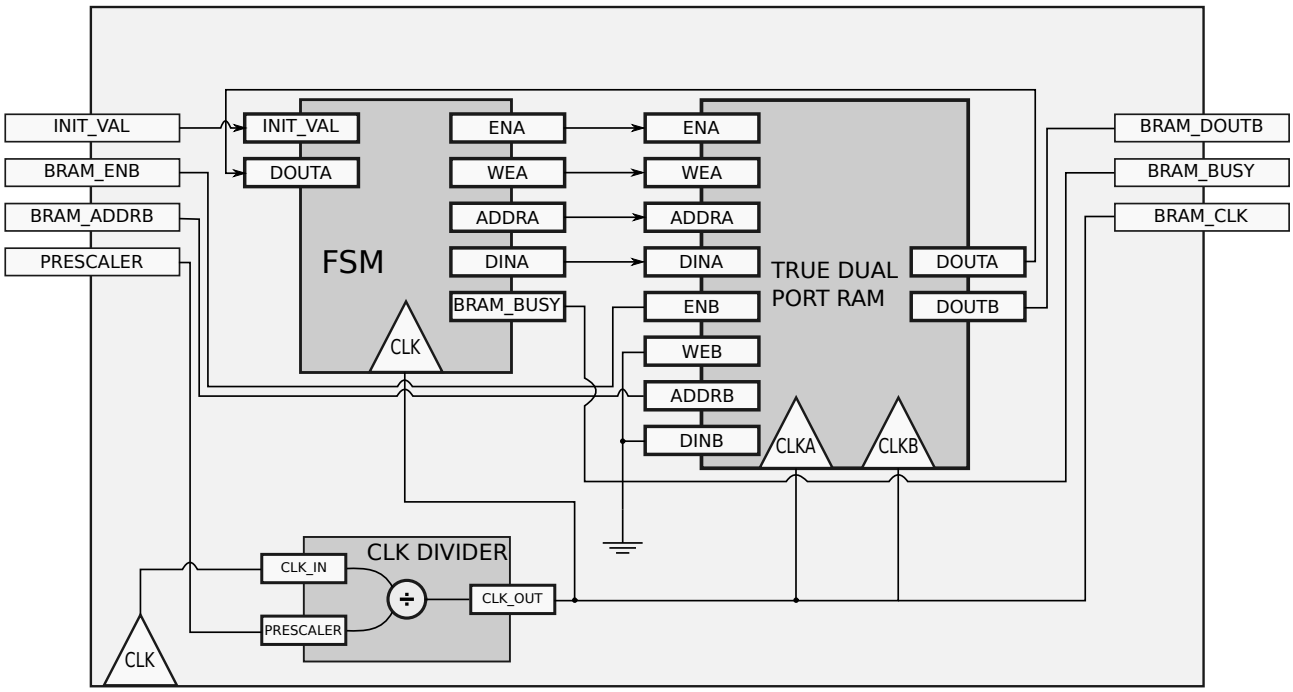

Figure 3.4: Adder Architecture