Watch Dog

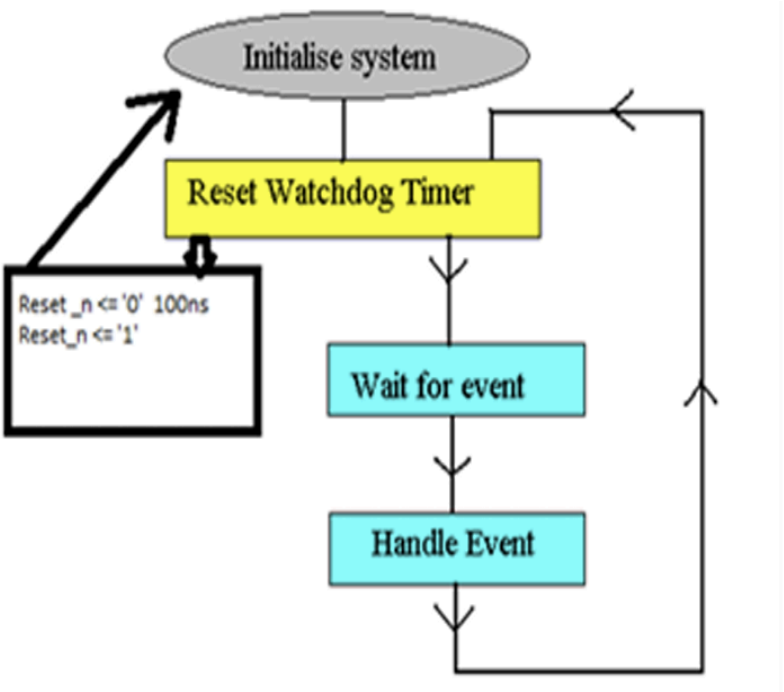
Introduction

In modern embedded systems, reliability and continuous operation are essential requirements. A common failure scenario occurs when software enters an unexpected state or becomes trapped in an infinite loop, causing the system to stop responding. To protect against such conditions, many designs incorporate a watchdog mechanism, which monitors system activity and triggers an automatic reset if the software fails to operate normally within a defined time window.

This project demonstrates the implementation of such a watchdog solution using a combination of hardware (VHDL) and software (C) on a MAX10-based FPGA platform. The hardware component implements a watchdog timer with a configurable timeout period, while the software periodically restarts the watchdog during normal execution. If the program intentionally or unintentionally fails to refresh the watchdog, the hardware generates a reset pulse, automatically restarting the system and restoring normal functionality.

The project provides a practical example of HW/SW co-design for improving system robustness and showcases how reset handling, timing mechanisms, and I/O control can be integrated within an embedded platform.

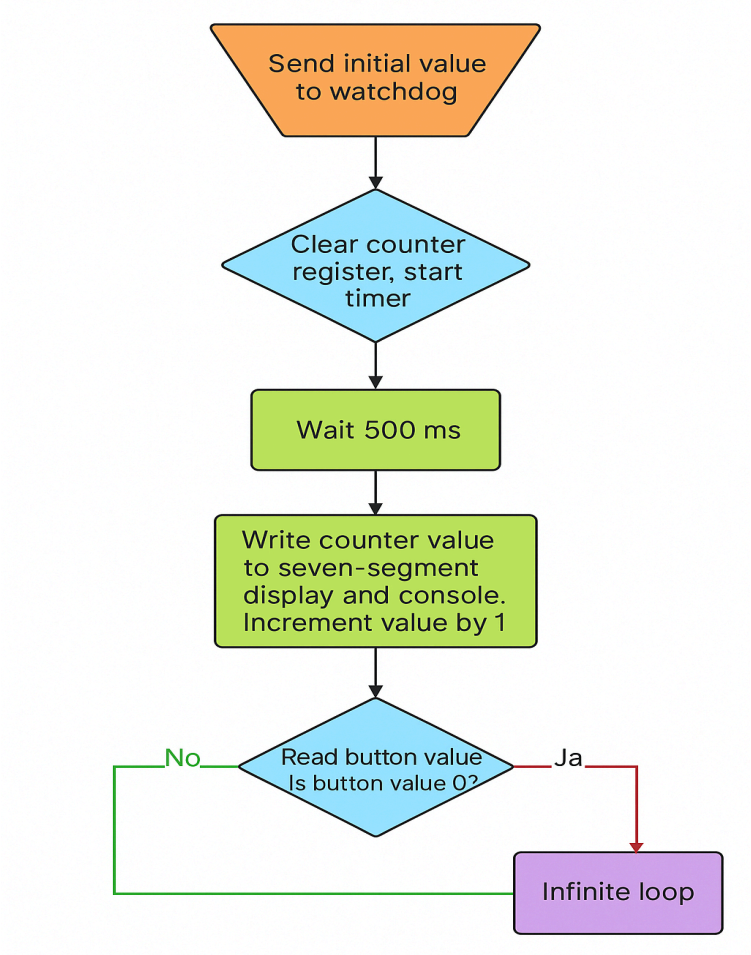
HW arkitektur av Watchdog Timer



SW architectur

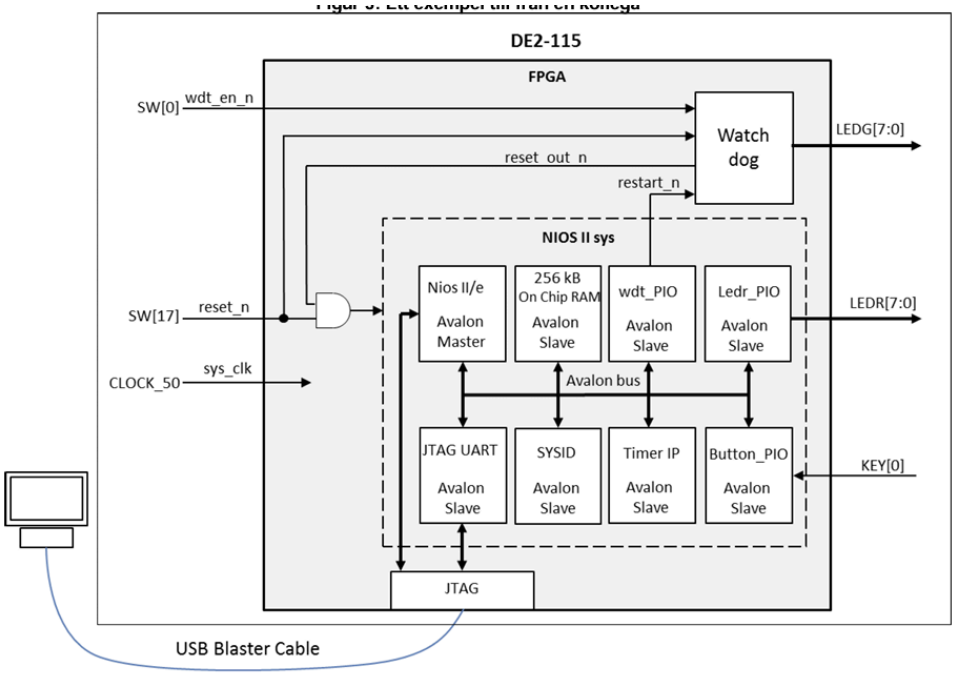
Referenser

Barr, M. (2001). Introduction to Watchdog Timers, https://barrgroup.com/embedded-systems/how-to/watchdog-timer (Only if you are so interested) Maxim Integrated Products, Inc. (2005, December). Supervisory Circuits with Windowed (Min/Max) Watchdog and Manual Reset, https://datasheets.maximintegrated.com/en/ds/MAX6323-MAX6324.pdf (Only if you are so interested) Texas Instruments, Inc. (2006). MSP430x1xx Family User's Guide, http://focus.ti.com/lit/ug/slau049f/slau049f.pdf



Example of a flow chart

```
#include <stdio.h> // device driver for printf
#include <altera_avalon_timer_regs.h> // device driver for my_timer
int main(void){
Printf("start program");
TIMER_RESET;
TIMER_START;
while(1)
{
//Start every 100ms//
while (TIMER_READ < 5000000){}; // wait 100 ms,
TIMER_RESET;
TIMER_START;
//Do some work, e.g. write a dot every second//
//Add the error code, a loop forever if KEY is pressed down//
//Kick the Watch_dog Timer//
}
```



system architecture

Demonstration Video – Watchdog System in Action

The following video shows the system running on the MAX10 board and demonstrates how the watchdog automatically resets the CPU when the program enters a fault state.

https://www.youtube.com/shorts/MmHCL7cK1l8