

Computer Architecture Lab 3

# Clock-HCS12

im Studiengang Technische Informatik  
der Fakultät Informationstechnik

Wintersemester 2022

Nikita Teztlaff, Fabian Zaiser

January 20, 2023

# Contents

|  | page     |
|--|----------|
| <b>1 Readme</b>                            | <b>1</b> |
| 1.1 General . . . . .                      | 1        |
| 1.2 Work distribution . . . . .            | 1        |
| <b>2 Functional requirements</b>           | <b>2</b> |
| 2.1 Software requirements . . . . .        | 2        |
| <b>3 User Interface of the program</b>     | <b>3</b> |
| 3.1 LCD Display . . . . .                  | 3        |
| 3.2 LED display . . . . .                  | 4        |
| 3.3 Operating buttons . . . . .            | 4        |
| <b>4 Module overview</b>                   | <b>5</b> |
| <b>5 Data dictionary</b>                   | <b>6</b> |
| 5.1 List of all global variables . . . . . | 6        |
| 5.2 Hardware resources . . . . .           | 7        |
| <b>6 Flowcharts of all modules</b>         | <b>8</b> |
| 6.1 main.c . . . . .                       | 8        |
| 6.2 clock . . . . .                        | 9        |
| 6.2.1 tick10ms . . . . .                   | 9        |
| 6.2.2 processEventsClock . . . . .         | 9        |
| 6.2.3 setClock . . . . .                   | 10       |
| 6.2.4 displayTimeClock . . . . .           | 11       |
| 6.2.5 setTimeZone . . . . .                | 11       |
| 6.3 button . . . . .                       | 12       |
| 6.3.1 checkButton . . . . .                | 12       |
| 6.3.2 buttonAction . . . . .               | 12       |
| 6.4 DCF77 . . . . .                        | 13       |
| 6.4.1 sampleSignalDCF77 . . . . .          | 13       |
| 6.4.2 compareLastSignals . . . . .         | 14       |
| 6.4.3 processEventsDCF77 . . . . .         | 15       |
| 6.4.4 DecodeTransmission . . . . .         | 16       |
| 6.4.5 checkParity . . . . .                | 17       |

# 1 Readme

## 1.1 General

The lab was worked on by **Nikita Tetzlaff** and **Fabian Zaiser**. It was examined by Profs Keller and Zimmermann on the 17.01.2023. The problems which occurred on the hardware were fixed with some inspiration by Prof Keller and the program is now working as described in the requirements chapter. The program was tested on 18.01.2023 and was now running on the hardware in the lab as it was intended.

## 1.2 Work distribution

### ➤ **Nikita Tetzlaff:**

- sampleDCF77() function
- buttonAction() function
- setClock() function
- flowcharts
- global variables list

### ➤ **Fabian Zaiser:**

- processDCF77() function
- DecodeTransmission() function
- checkParity() function
- checkButton() function
- tick10ms() function modified
- ifdefs for automatic toggling between simulation and on hardware testing so no code has to be modified when switching between simulation and testing on hardware
- assembled the documentation in LaTeX

## 2 Functional requirements

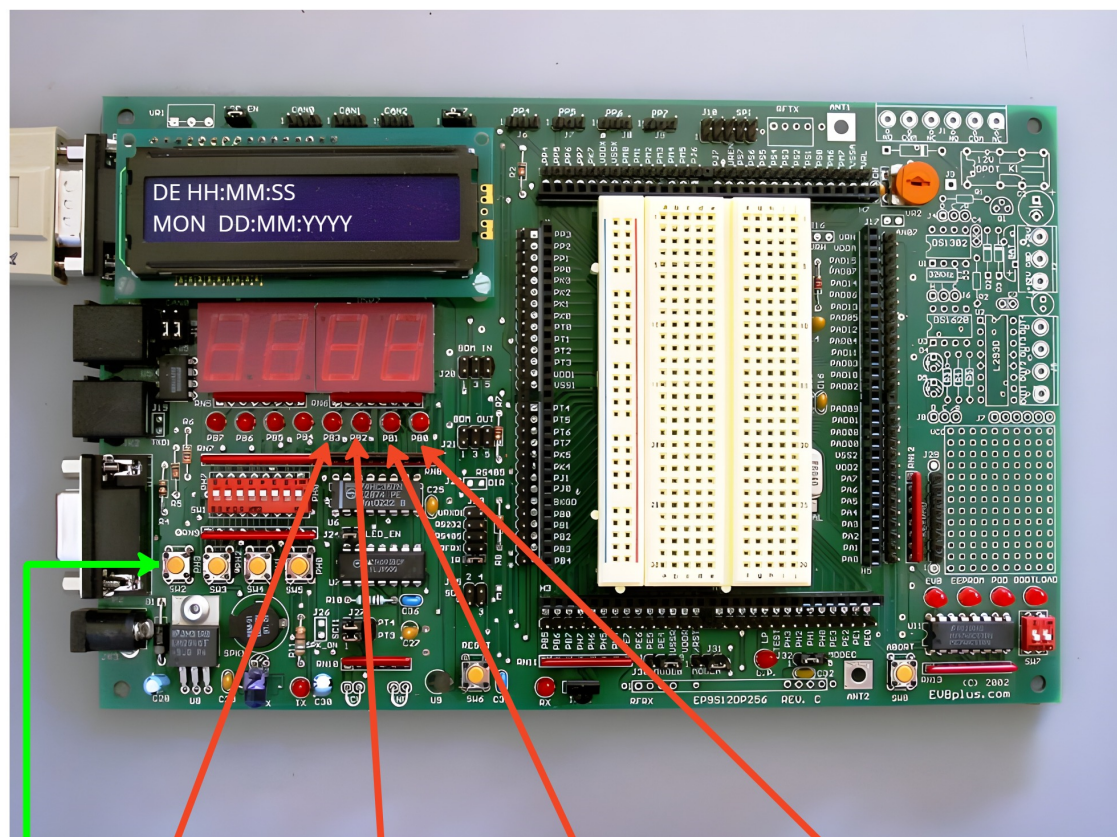
### 2.1 Software requirements

- The first line of the Board shall display the time in the format hours: minutes: seconds
- The second line shall display the date in the format day .month .year
- The clock shall continue working as in the lab before when no radio transmission is received or it is failed
- As in the previous lab the B.0 LED shall toggle every second
- The B.1 LED shall be turned on when the signal of the DCF77 is low and shall be turned off when the Signal is high
- The LED on B.3 shall light up when the time and date has been decoded. It is turned off if the transmission fails.
- the DCF77 signal shall be polled every 10ms
- the Weekday MON(day), TUE(sday), ... shall be displayed infront of the date.
- pressing the H3 button shall toggle between European Time and US eastern timezone (European time - 6hrs)
- The displayed timezone shall be indicated by either "DE" or "US" on the LCD

## 3 User Interface of the program

### 3.1 LCD Display

- 1st line: timezone and time display
- 2nd line: weekday and date display



- B.0 toggles every second
- B.1 on when signal of DCF77 low/ off when high
- B.2 lightsup when an error occurred
- B.3 on when decoding successful/ off when transmission failed
- H.3 button to toggle between DE and US time

miro

Figure 3.1: DRAGON-12 EVALUATION BOARD

## **3.2 LED display**

- LED0: toggles once per second
- LED1: on when signal is low and off when signal is high
- LED2: is on when transmission of the signal has failed
- LED3: lights up when decoding was successful

## **3.3 Operating buttons**

- SW2 (H.3): pressing H.3 toggles between the timezones

## 4 Module overview

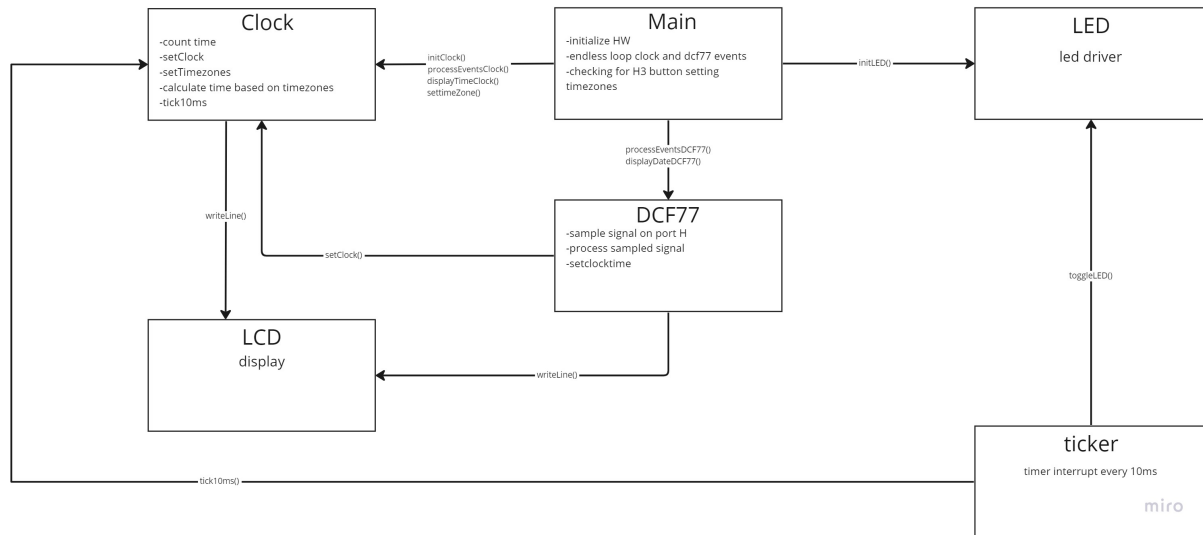


Figure 4.1: overview over all modules

# 5 Data dictionary

## 5.1 List of all global variables

| Module where variable is declared | Variable name       | C like datatype | Purpose   |
|-----------------------------------|---------------------|-----------------|---|
| clock                             | clockEvent          | CLOCKEVENT      | Variable holding the last clock event   |
|                                   | hours               | static char     | Hours (as numbers)  |
|                                   | minutes             | static char     | Minutes (as numbers)  |
|                                   | seconds             | static char     | Seconds (as numbers)  |
|                                   | uptime              | static int      | software timer  |
|                                   | ticks               | static int      | ticks used to calculate elapsed time  |
|                                   | timezone_curr       | static int      | current timezone flag   |
|                                   | timezoneChanged     | extern int      | flag if timezone has been changed since last setClock                             |
| dcf77                             | dcf77Event          | DCF77EVENT      | Variable holding the last DCF77event  |
|                                   | dcf77Year           | int             | decoded year based on DCF77-signal  |
|                                   | dcf77Month          | int             | decoded month based on DCF77-signal   |
|                                   | dcf77Weekday        | int             | decoded Weekday based on DCF77-signal   |
|                                   | dcf77Day            | int             | decoded day based on DCF77-signal   |
|                                   | dcf77Hour           | int             | decoded hour based on DCF77-signal  |
|                                   | dcf77Minute         | int             | decoded minute based on DCF77-signal  |
|                                   | weekdayArray        | static char*    | Array holding the shorthand symbols for weekdays                                  |
|                                   | lastSignals         | static int [10] | array storing the bits of the last signals (needed to eliminate spikes in signal) |
|                                   | stateMachineEnabled | int             | Blocks processing of event while dcf77Event is invalid                            |
|                                   | transmission        | int [59]        | array to store the transmissioned signal and used to check parity                 |
|                                   | transmIndex         | int             | index to walk over array transmission   |
|                                   | stateMachineEnabled | int             | Blocks processing of event while dcf77Event is invalid                            |

Figure 5.1: list of global variables



## 5.2 Hardware resources

| Module                | HCS12 or Dragon12 HW resource       | Purpose   |
|-----------------------|-------------------------------------|---|
| Clock                 | CPU Port K<br>LCD Display           | Display: Line 1; display time<br>Line 2: display date |
| LED<br>[clock][DCF77] | Port B<br>LEDs                      | Various status signals                                |
| Clock<br>[buttons]    | Port H.3<br>Buttons SW2             | Toggle european time<br>and US east time              |
| Ticker<br>[clock]     | Enhanced Capture Timer<br>Channel 4 | 10 ms ticker  |

Figure 5.2: list of hardware resources

# 6 Flowcharts of all modules

## 6.1 main.c

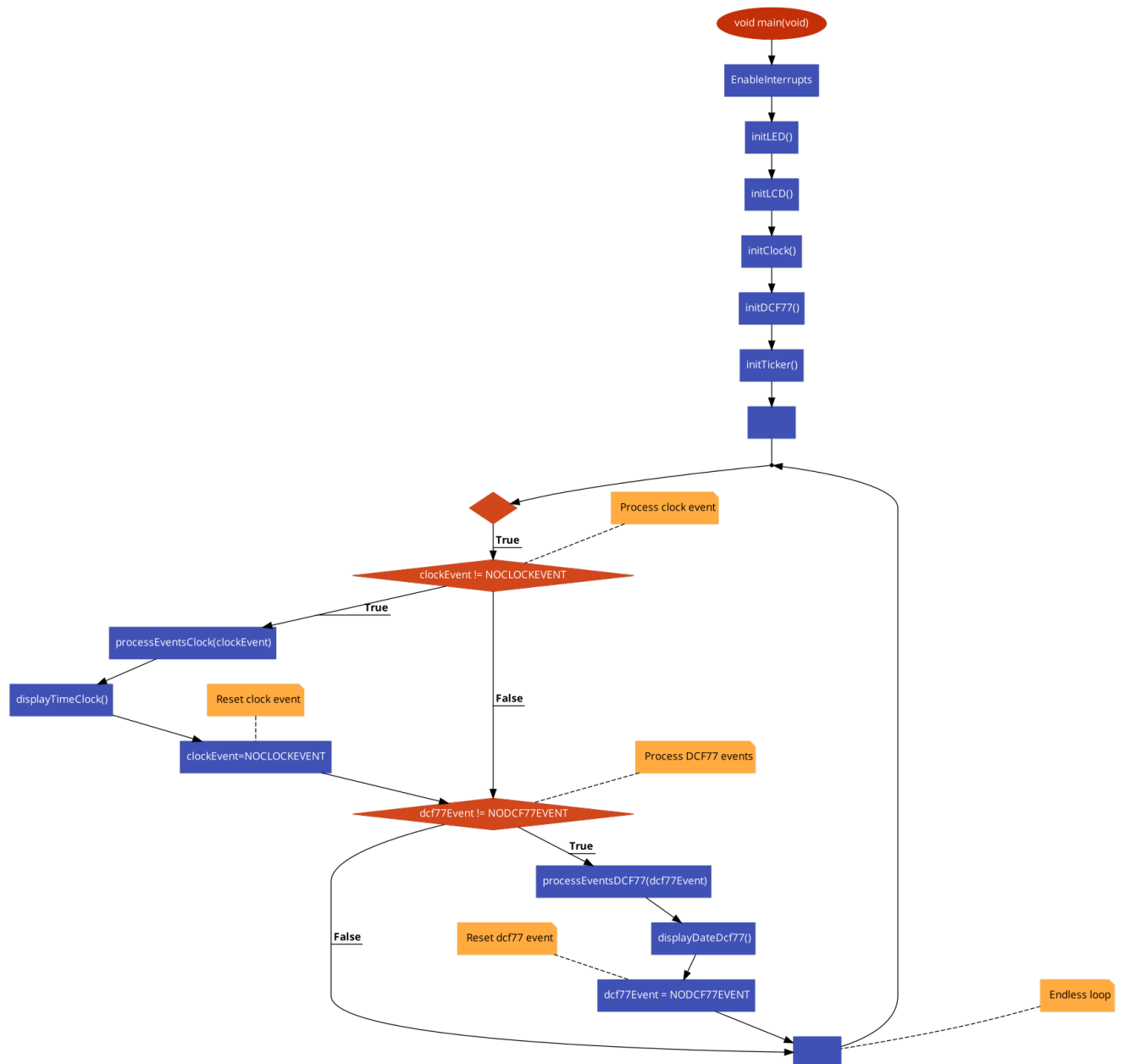


Figure 6.1: main() flowchart

## 6.2 clock

### 6.2.1 tick10ms

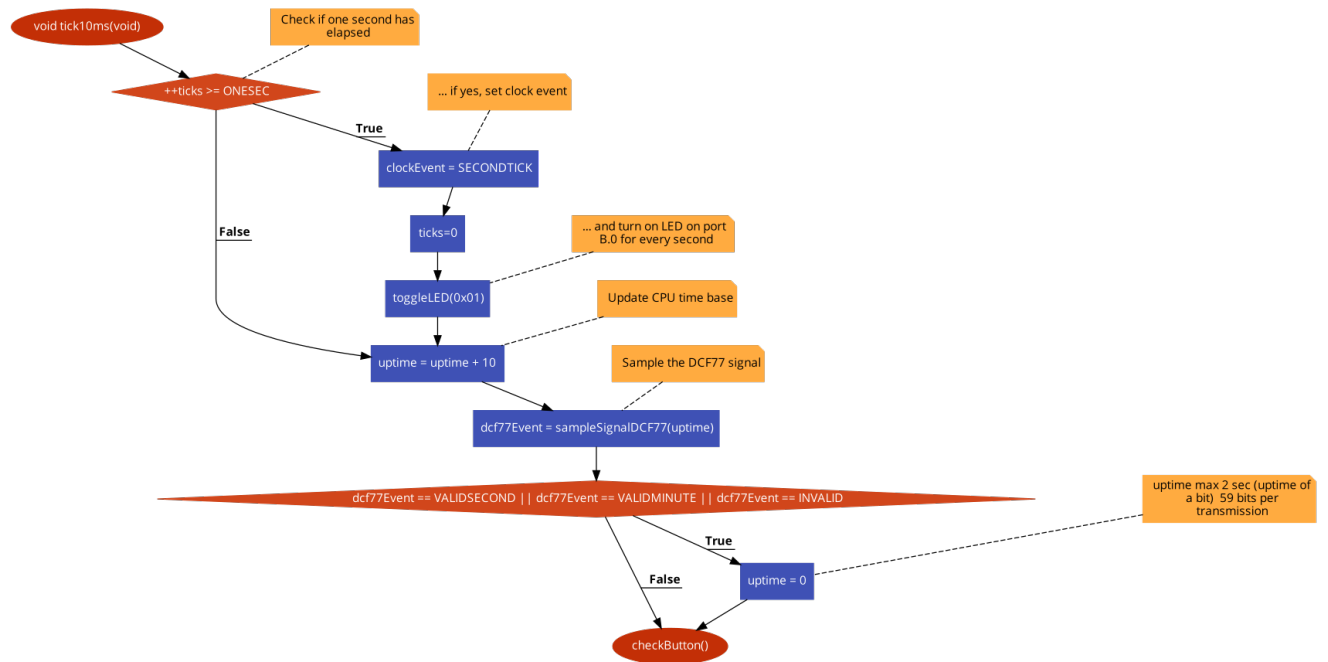


Figure 6.2: tick10ms flowchart

### 6.2.2 processEventsClock



Figure 6.3: processEventsClock flowchart

## 6.2.3 setClock

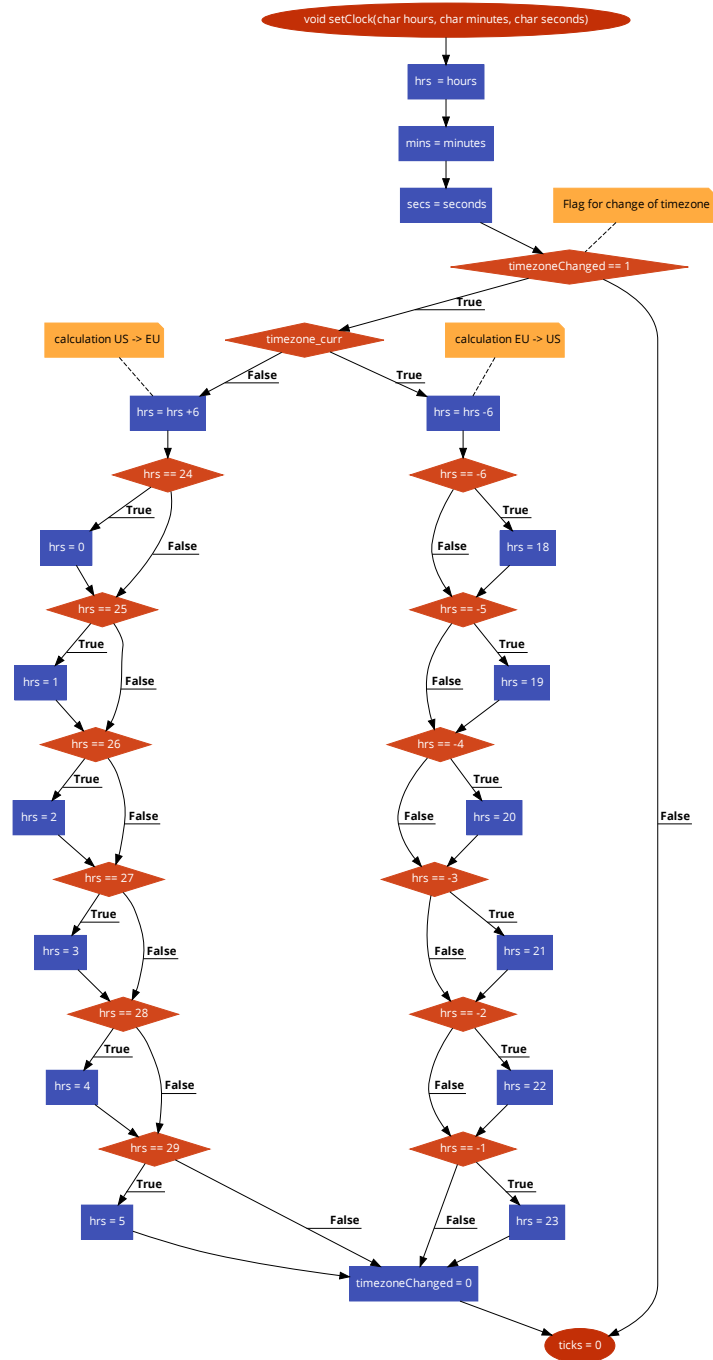


Figure 6.4: setClock flowchart

## 6.2.4 displayTimeClock

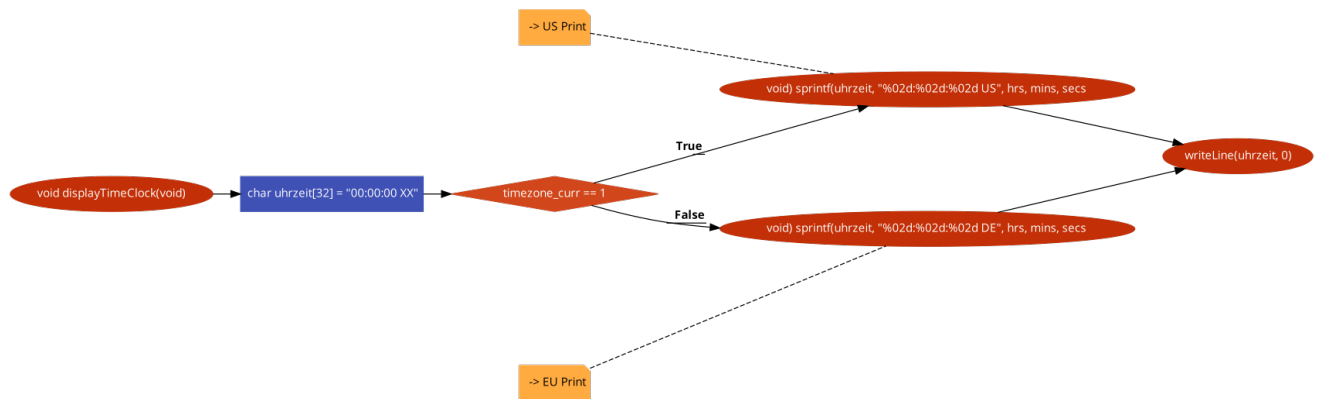


Figure 6.5: displayTimeClock flowchart

## 6.2.5 setTimeZone

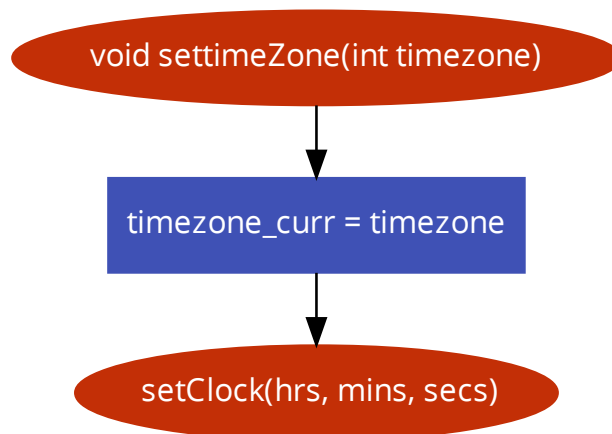


Figure 6.6: setTimeZone flowchart

## 6.3 button

### 6.3.1 checkButton

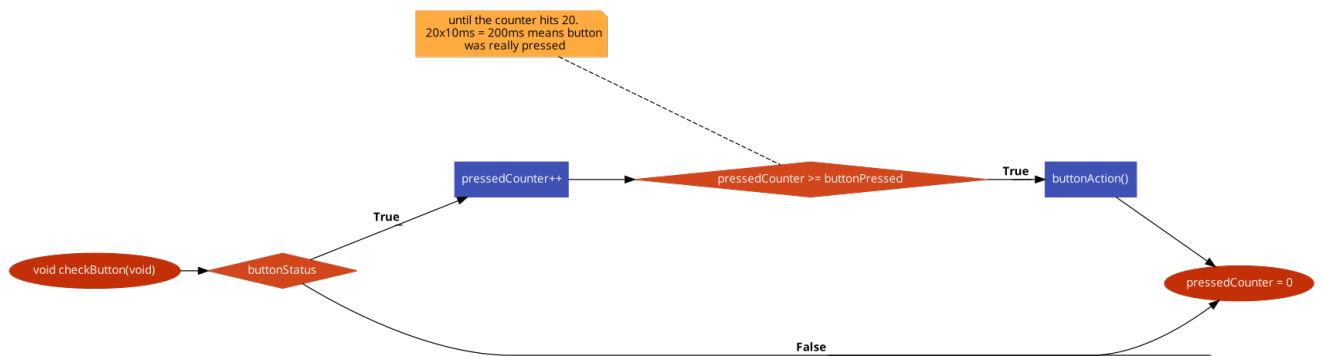


Figure 6.7: checkButton flowchart

### 6.3.2 buttonAction

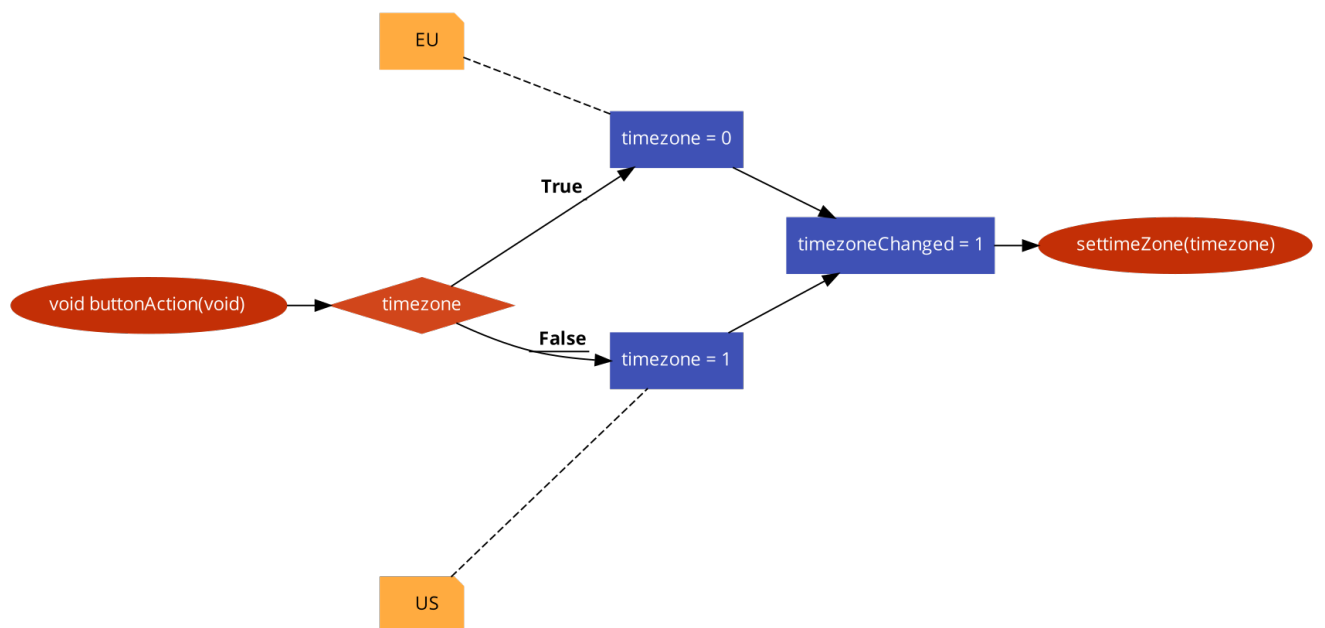


Figure 6.8: buttonAction flowchart

## 6.4 DCF77

### 6.4.1 sampleSignalDCF77

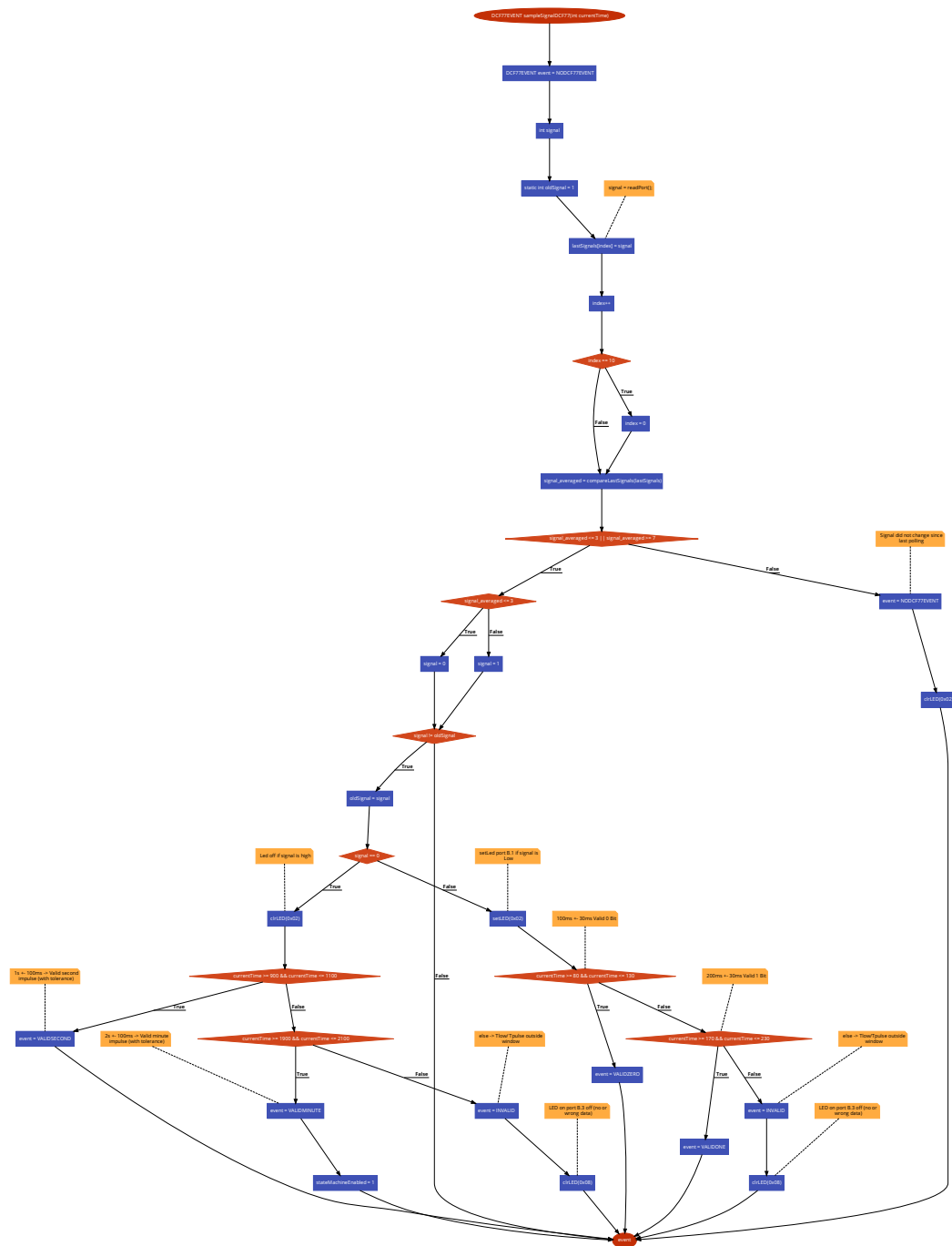


Figure 6.9: sampleSignalDCF77 flowchart

### 6.4.2 compareLastSignals

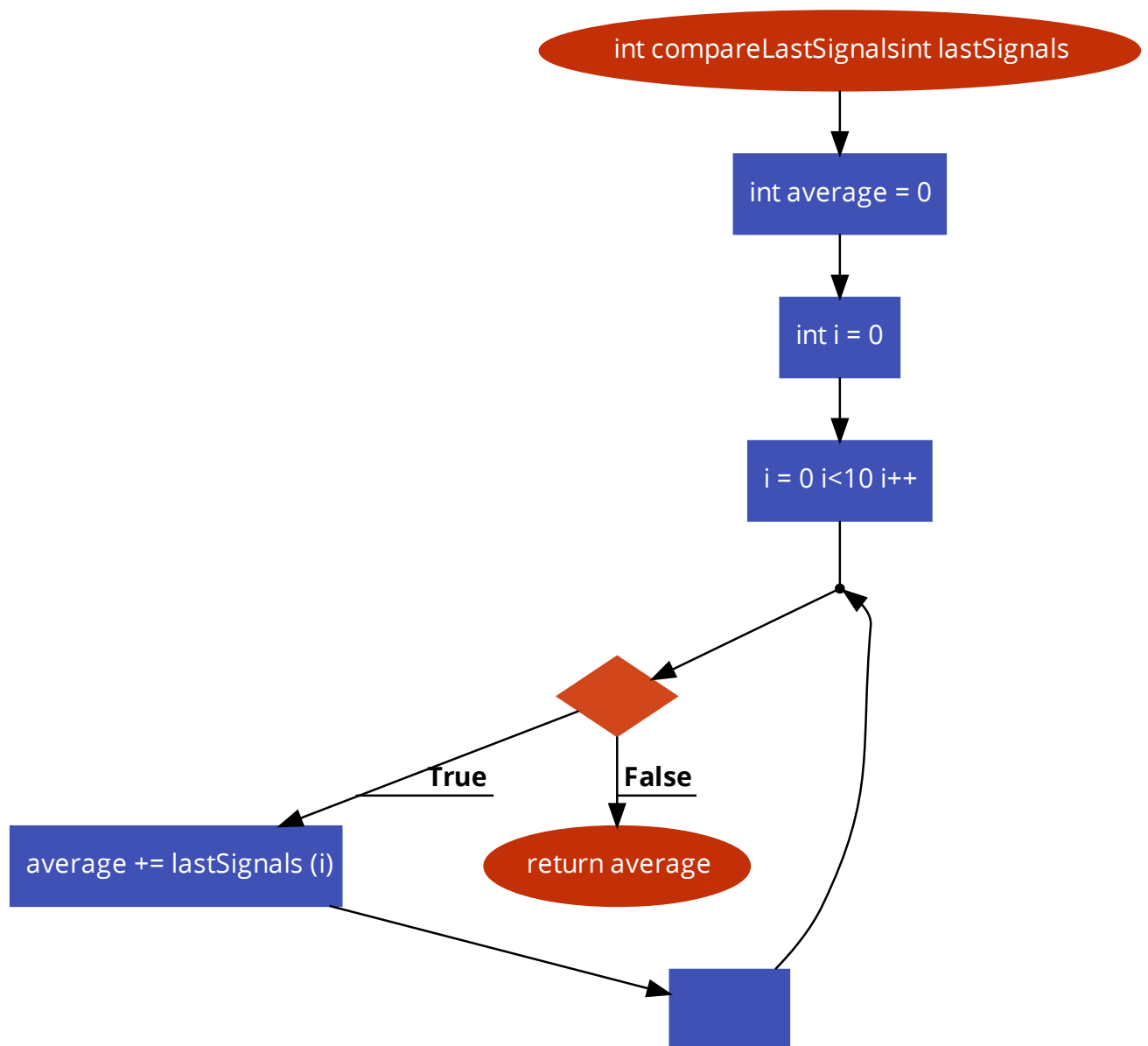


Figure 6.10: compareLastSignals flowchart



### 6.4.3 processEventsDCF77

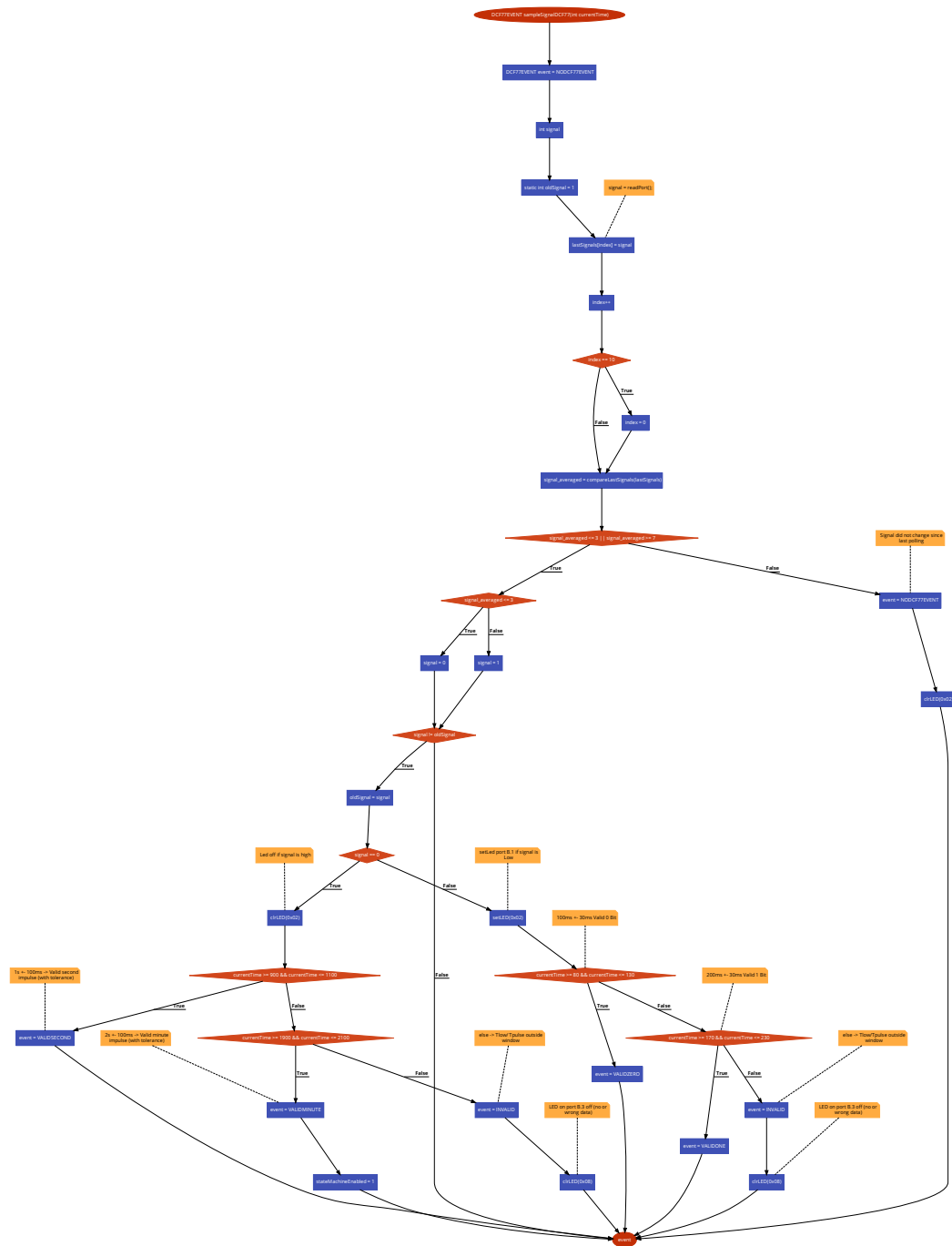


Figure 6.11: processEventsDCF77 flowchart

#### 6.4.4 DecodeTransmission

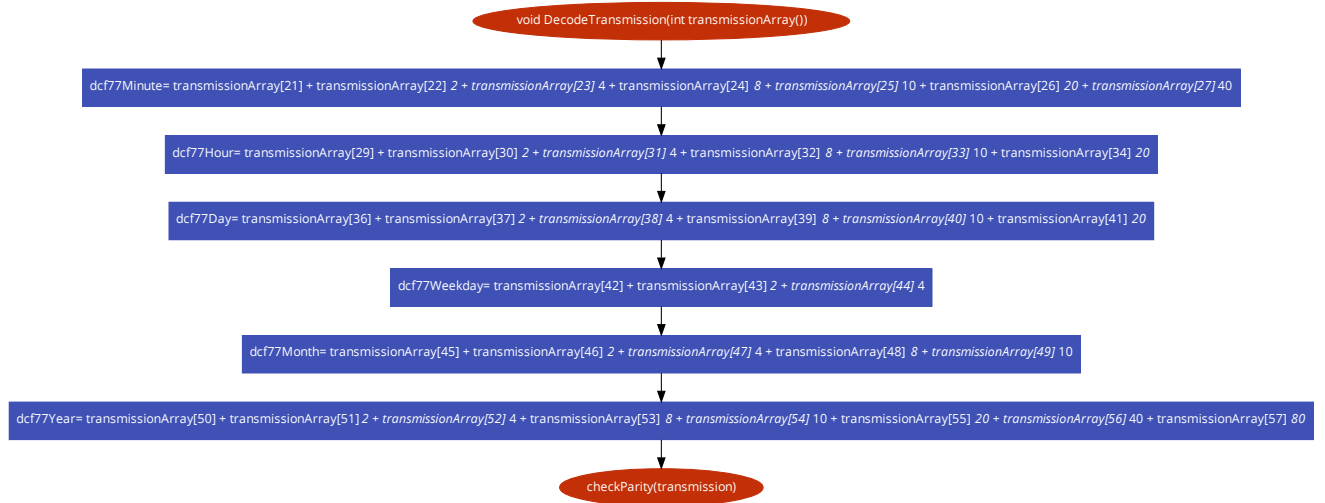


Figure 6.12: DecodeTransmission flowchart

### 6.4.5 checkParity

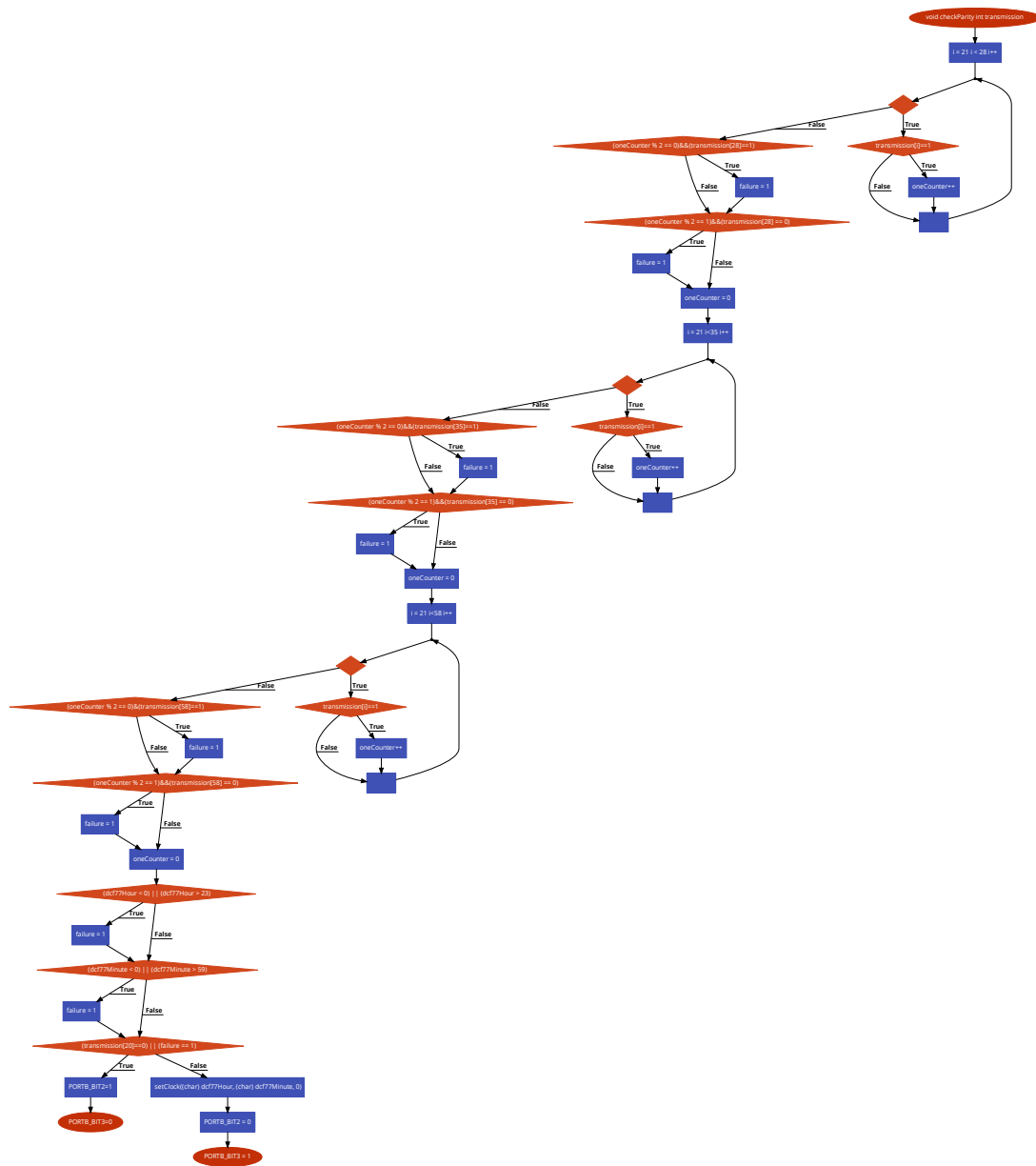


Figure 6.13: checkParity flowchart