

ELEVATOR'S CONTROL SYSTEM BASED ON XC 4000XL FAMILY OF XILINX FPGA

Albert Fazakas, Sorin Hintea, Mihaela Cîrlugea, Dan Petrescu

Technical University of Cluj-Napoca, 26-28 Baritiu str. , Albert.Fazakas@bel.utcluj.ro,
Sorin.Hintea@bel.utcluj.ro, Mihaela.Cirlugea@bel.utcluj.ro

Abstract: This paper describes how to design an elevator's control system with two chambers using FPGA circuits. The control system memorizes and controls calls received from different floors and drive the chambers to reach the calling point in shortest time possible. The control system preserves an efficient user interface, accomplished by displays (clock, calls, chamber's position) and this will give real monitor facilities (the current speed and actual position of the chambers, sensor's functioning, damage and alarm) and possibilities of technical support to the supervisor in the hot points of the system function. The modular structure of the project and software solutions of the XILINX FOUNDATION package allows changes in controller functions with small corrections in the hardware.

Keywords: elevator, FPGA circuits, programmable device, VHDL language

1. INTRODUCTION

In this paper is offered the model design work of a control system of an two chambers elevator in XILINX technology using a FPGA device fore one-chip approach. This system contains many facilities, especially using the environment XILINX FOUNDATION which offers a great design flexibility. Because of using FPGA circuits, any behavioral change can be easily done by reediting the electrical circuit diagram or the VHDL instructions that describe the corresponding modules. We have the goal of completing a complex project that can be implemented using a Xilinx component that has small dimensions, XC4013.

The control system memorizes and controls calls received from different floors and drives the chambers to reach the calling point in shortest time possible. There is a priority system for the calls. For example, if somebody is in the chamber, the higher priority will have the call coming from the inside. If nobody is in the chamber the system will wait for external calls. A chamber will execute all the valid calls from its direction being continuously able to receive other valid calls, the system recounting its route. If a chamber is not used for minimum one hour and maximum two hours, the control system will bring it into the Stand By position and then it will wait the external calls unless the HOLD signal is not activated. The system has its own security: each chamber has weight limits and possibilities of controlling the speed. Obviously, the start is possible only with closed doors. In special cases like fire alarm, which can be

activated by the supervisor or by the fire security system of the building, the chambers will stop at the nearest floor and the doors will open. When the chambers are free, they will move towards the Stand By position. When both chambers are in Stand By position the control system will send at a certain hour (for example at seven o'clock) one chamber to the ground floor and the other to the sixth floor, in order to prepare the chambers positions for high-traffic periods.

2. SYSTEM DESCRIPTION

The controlled elevator has two chambers that serve for six stages, ground floor and the Stand-By level, so there result eight different positions. On every stage/floor there are for each chamber a couple of control buttons and a display with eight positions that shows the user the location of the respective chamber. Inside each chamber there is a panel with nine buttons, seven for inside calls, one for canceling inside calls (marked with *zero* symbol) and one for alarm and calling the system supervisor. The supervisor can access other four supplementary buttons: SB (Stand-By), HOLD, CLOSE_D (CLOSE DOOR) și OPEN_D (OPEN DOOR). Inside each chamber there also exists a display with eight positions alike the display from each floor that describes the current position of the chamber (Red) and the valid called positions of the chamber (Green).

The control system assures following preliminary functions:

All valid calls are memorized and managed by the system. A valid call is signaled by the control button properly lightened and can be done by that chamber in it's present state. For example, if a chamber descends and it reaches the IVth floor, a DESCEND call from every floor on the moving direction (III, II, I, GND) will be valid, while a call from the fifth floor or an ASCEND call will not be valid. This idea wants to eliminate conflicts in managing the calls. Obvious, a security condition is, that start is possible only with closed doors. Doors close automatic, if the sensors do not detect a person for 4 seconds.

The system is equipped with a manage algorithm of call priorities.

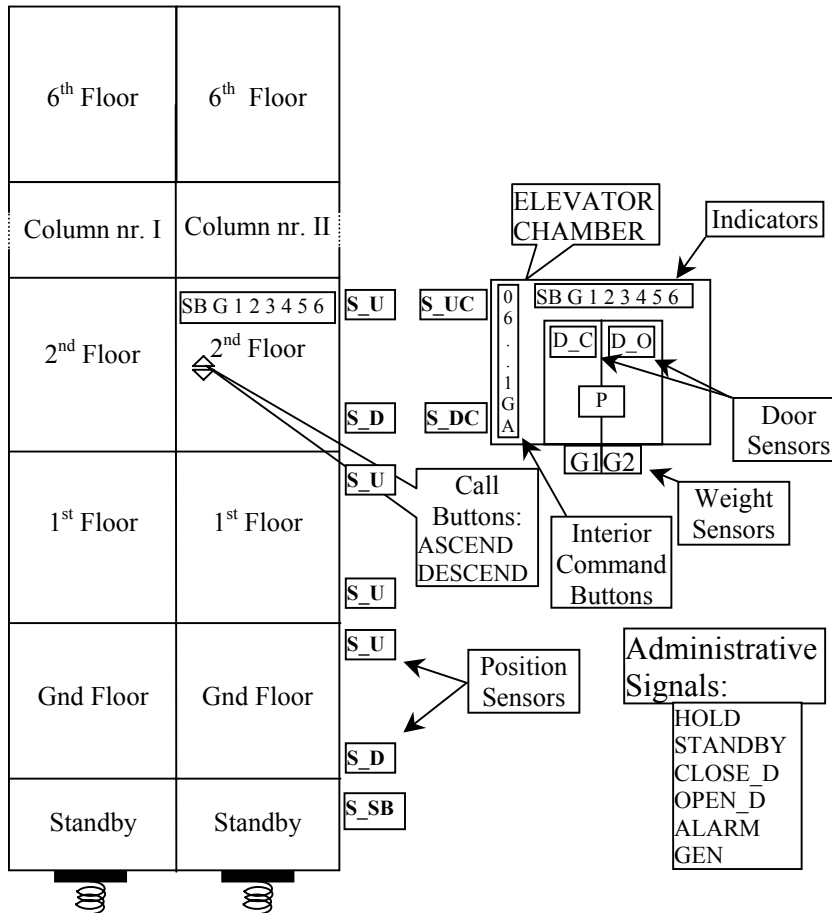
The facility and security measures of the system are:

- modifying possibility of an external call: if, for example, an user calls a chamber with an external call DESCEND and, after entering the chamber, he makes a call for ascending, the system will execute the call, without any conflict.
- weight supervision inside the chamber, with two limits (Chamber empty – 20kg and Overweight – 300Kg)
- speed supervision for each chamber (with a high and a low limit)
- Stand-By position is provided with system force protection and with technical maintenance
 - at certain hours (for example at seven o'clock), if both chambers are in Stand-By, the control system will send one chamber to the ground floor and the other to the sixth floor, in order to prepare the elevator system for high-traffic periods
 - in special cases like fire ALARM, which can be activated by the supervisor or by the fire security system of the building, the chambers will stop at the nearest floor and the doors will open. Then, the system gives an acoustic signal. When chambers are free, they will move towards the Stand By position
 - the sensor's signals have a control circuit that monitorizes the function of each. Their state is displayed for the supervisor
- the system is provided with decisions for power failure and switching to an auxiliary

power source cases

3. DESCRIPTION OF SYSTEM DESIGN IN A BLOCK STRUCTURE

One describes the system structure in a block structure. For an optimal design and a good flexibility, one choosed a modular structure grouped on levels, depending on



the blocks position according to the interface of the controlled assembly.

Figure 1. Structure of the implemented system

3.1. Input-output signals

The communication between the designed control system and the controlled assembly is obtained from the primary input-output signals. In figure no.1 one can see the columns section and the section of one chamber with the corresponding primary input signals.

Primary input signals can be divided into two cathegories:

- Controls:

External call: from external control buttons (CB) UP respective DOWN (6+6);

Internal call: from internal control buttons (ICB) (7), a cancel signal (0) and an “A”;

Other controls: there are 4 Administrative Signals (AS) (HOLD, SB, CLOSE_D, OPEN_D) and have an andministrative purpose

- Sensors:

Weight: there are 2 weight sensors based on the weight sensors under the floor:

G1 (for $G > 20$ kg) and G2 (for $G > 300$ kg);

Position: come from the position sensors and there are two sensors a level: S_U and S_D, respective two for each chamber: S_UC and S_DC. Supplementary, there is another sensor for the Stand-By position, S_SB. These signals are used for determining the position and measuring the speed of the elevator.

Door: come from the sensors corresponding to the doors: D_C (= door is closed), D_O (= door is open), P (=people coming to the door);

ALARM: comes from the security system of the building or from the supervisor button

GEN: comes from the control system supply in case of powerfail and switching to the additionally generator

Primary output signals are shown in figure 2 and are of 3 types:

- Controls: (used for the power engine): UP; DN; BREAK; STOP.
- Action: (for the door engine): CLOSE ; OPEN.

- Interface: (shown in the block USER INTERFACE): C_P – current position (8) – for internal display (red) and external; S_P stop_positions (7) – for internal display (green); LED_U (6) – lightening of valid control buttons UP (the call was accepted for them and the chamber will stop at that floor); LED_D (6) – lightening of valid control buttons DOWN; MES/BEEP (14) – for message block; ADMIN (19) – for the supervisor display; C_TIME (42) – for displaying Current Time inside the chambers.

For the signals described above, the denomination tells their role, but further details of their effects and behaviour is not the purpose of the present paper.

There are not only primary but also secondary signals that establish relations between the blocks. These signals are internal and they are optimized, modified or debugged by the component blocks.

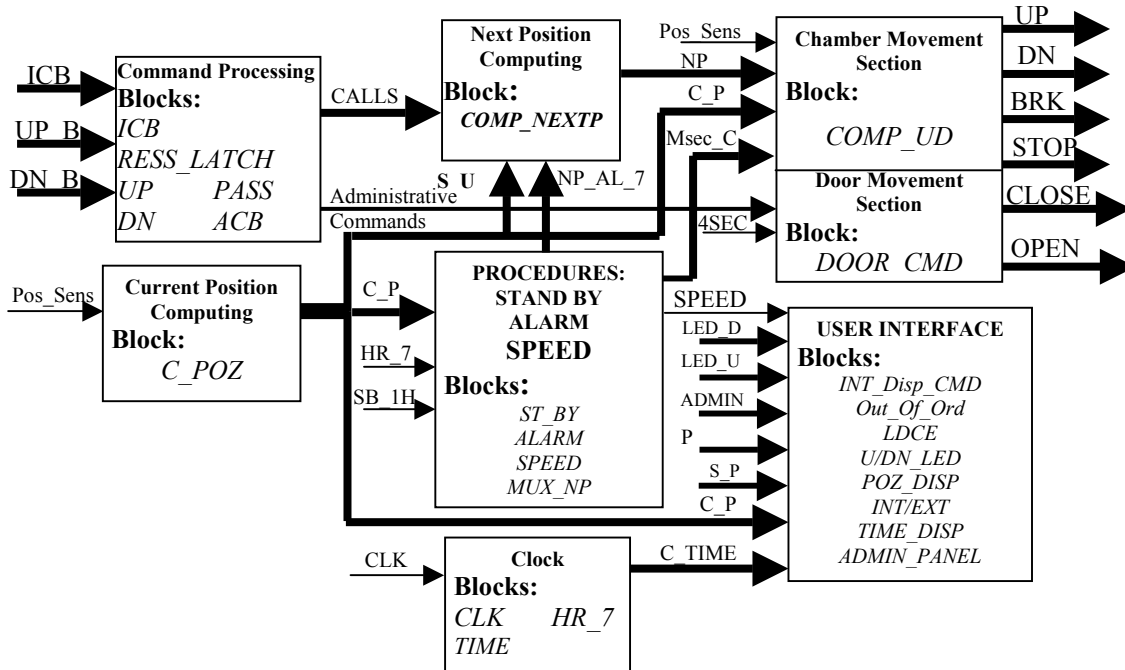


Figure 2. Block diagram and the block's content

3.2. The block diagram

The block diagram in figure 2 has a modular structure, hierarchized on circular concentric levels. Here are the functions of these levels and their composition:

Level I is a signal assembly that communicates with the user and with the controlled system, thus the primary signals

Level II contains blocks that serve level III and blocks that generate primary signals in final shape:

- Input blocks: ICB; DN; UP; ACB (used to process the input commands); C_POZ (for calculating the current position); SPEED; TIME; CLOCK; HR_7; ALARM; MUX_NP (used to recalculate the route of a chamber in movement)

- Output Blocks: The blocks contained in the User Interface Section and the blocks: COMP_UD, DOOR_CMD.

Level III (CORE) is the core of the control system and contains blocks that determine the signal behaviour, signals that control principal actions of the system. The blocks are: COMP_NEXTP (to compute the next position of a chamber); ST_B (used for watching the standby conditions – for instance, one hour of inactivity); RESET.

4. DESIGN IMPLEMENTATION

The design was implemented using Xilinx Foundation 2.1 Design environment. The design was based both on schematic and VHDL approach. In this way, it is easy to take advantage from both approaches: schematic approach helps us for organizing hierarchical design into modules, while the VHDL language is very suitable to design the complex circuit functions – needed for the various elements in this structure.

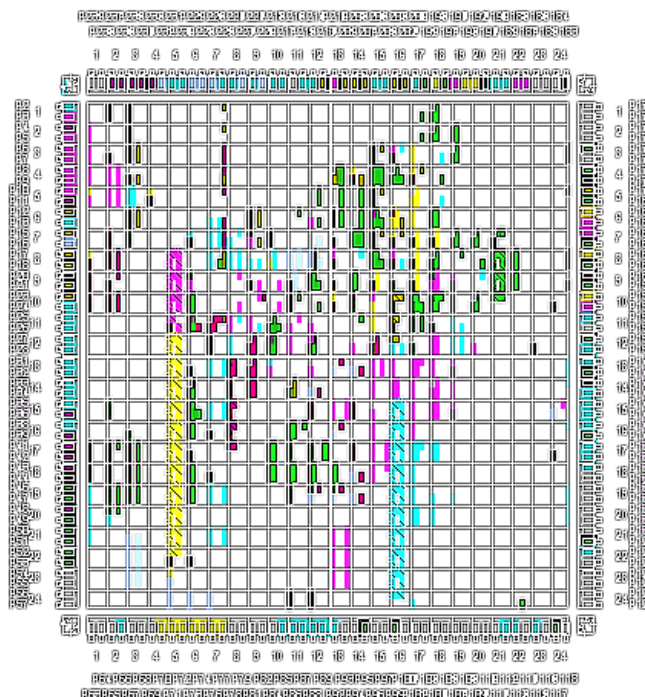


Figure 3. Floorplan of the target device XC4013

The target device is XC4013XL in the PQ240 package, from the Xilinx XC400 family. The implementation report shows us a number of 271 CLBs (Complex Logic Blocks) out of 576, which means an occupation of 47%. The total equivalent gate count for the design is 4393 and 6576 additional JTAG gate count for the IOBs (Input-Output

Blocks). The only reason why the design did not fit in a smaller device (like XC4010), is the high number of input-output signals needed by the the system, which cannot be provided with a XC4010 device (in the package PC84 – 84 pins).

Figure 3 shows the floorplan of the target device. The white squares represents unused CLBs. The maximum net timig delay in the chip is 11ns and the maximum circuit timing delay is 80ns, which means that the design can operate at frequencies of tenth of MHZ, however the designed system does not need a high speed response.

5. CONCLUSION

The example shown in this paper suggest us that FPGA circuits, although widely used for high-speed designs, can be used for lower-speed designs too, and due to their flexibility, they can be a very good alternative for microcontroller-based designs.

Facilitatile obtained by the design environment XILINX FOUNDATION gives a special design flexibility. The programming environment assures the interface between high level design and chip design. Another facility is the possibility of logic simulation and removing the design errors. The program creates the documentation necessary for the project, containing the optimized implementation of logical equations and the correspondence between the integrated circuit pins and their signals. The variety of possible architectures helps the user in choosing the proper circuit for his application. The project can be generally implemented by using the standard prgrammable circuits. In this particular case, the design flexibility is much more important because of the modular configuration of the designed circuit. Any behavioural modification can be easily done by reediting the electrical circuit or the VHDL code which describe the modules. One can also correct very easy possible design errors.

PLD performances are given by the response speed equal or less than the speed of discrete logic circuits and by the low power consumption. But, performances cannot be measured in power consumption because these circuits replace more discrete circuits which consumption can be bigger than that of a programable circuit. Most programable circuits are completed in CMOS technology, so they have zero consumption in “wait” mode.

Using programable circuits reduces costs by saving printed circuit space, reduces storage costs and protects the circuits against copying.

All these reasons contribute to increasing the use of logic programable circuits, because of simplifying the design methods and decreasing the implementation costs.

6. REFERENCES

- [1] Xilinx Foundation Series Software Manual. Xilinx Inc. (2000)
- [2] Hintea S. (1998) Tehnici de proiectare a circuitelor digitale VLSI. Editura Casa Cărții de Știință, Cluj-Napoca.
- [3] Cîrlugea Mihaela, Hintea S. (1999) Introducere in VHDL, Editura Casa Cărții de Știință, Cluj-Napoca.