www.editada.org

International Journal of Combinatorial Optimization Problems and Informatics, 18(1), 2027, 111-125. ISSN: 2007-1558. https://doi.org/10.61467/2007.1558.2027.v18i1.1508

A Traceable Design Flow for Synchronous FPGA Architectures: From State Machine Specification to Hardware Validation on the Alhambra II Platform

Danna Paola Lozada Padilla[1], Carlos Daniel Zamora Pérez[2], Romina María Rodriguez Garcia[1], Gilberto Ortega Méndez[2], Juan Salvador Mejía Salvador[1], Justo Fabian Montiel Hernández [3]

[1]Escuela de Ingeniería, Universidad La Salle Pachuca, México.

[2]Departamento de Ingeniería Eléctrica y Electrónica, Tecnológico Nacional de México, Campus Pachuca, México.

[3]Escuela Superior de Ciudad Sahagún, Universidad Autónoma del Estado de Hidalgo

232070@lasallep.mx, l22200629@pachuca.tecnm.mx, 232125@lasallep.mx, gilberto.om@pachuca.tecnm.mx, jmejia@lasallep.mx, justo_montiel@uaeh.edu.mx

Abstract: This paper presents a structured, reproducible methodology for the design and FPGA-based implementation of a synchronous digital clock on the open-source Alhambra II platform using IceStudio as the development environment. The proposed approach systematically integrates finite state machine specification, state transition and excitation table construction for T flip-flops, Karnaugh map minimisation, and Boolean equation synthesis, establishing a direct, traceable path from functional specification to physical hardware. The resulting modular architecture comprises a frequency divider, synchronous BCD counters for seconds, minutes, and hours, hierarchical carry propagation logic, and a display subsystem interfaced through 74LS48 decoders and six seven-segment displays. Experimental validation on the Alhambra II confirms full functional correctness and demonstrates the consistency between the formal logical model and the synthesised hardware. The primary contribution is a validated, complete design flow (from FSM specification to FPGA implementation) applicable as a scalable pedagogical and technical framework for embedded sequential digital systems based on open-source programmable logic devices.

Keywords: Finite state machine, Karnaugh map minimization, Open-source hardware, Reconfigurable logic device, Modular digital architecture, BCD counter, IceStudio, Embedded sequential systems, Boolean equation synthesis.

Article Info
Received: Jun 30, 2026
Accepted: Aug 01, 2026

1 Introduction

The increase in the complexity of embedded digital systems has driven the adoption of programmable logic devices, particularly Field-Programmable Gate Arrays (FPGAs), due to their capacity to implement digital architectures with high parallelism, configuration flexibility and reduced development times (Kilts, 2007), (Wolf, 2004). The possibility of reconfiguring the hardware without modifying the physical platform has favoured their incorporation in automation, instrumentation, digital signal processing, control systems and educational platforms for teaching digital design (Leong, 2008), (Simpson, 2010), (González, 2026). Several works have presented implementations of digital clocks on FPGA platforms using hardware description languages such as VHDL and Verilog, as well as synthesis tools oriented towards the development of digital systems (Navarro-Torrero et al., 2024), (Ashenden, 2007). Most of these proposals focus attention on the functionality of the system and on the description of the code, while a reduced number incorporate a detailed analysis of the modular architecture (López Monterroso, 2020), of the consumption of logic resources, of the temporal behaviour during synthesis and of experimental validation using physical hardware (Dillien, 2017). This situation limits the reproducibility of the design and the objective evaluation of its efficiency within different programmable platforms (Bolaños et al., 2023), (Cortez et al., 2023).

Consequently, the need is identified to develop implementations that, in addition to fulfilling the required functionality, allow the analysis of the modular organization of the system, the quantification of the utilization of resources available in the programmable device and the verification of the temporal behaviour obtained after the synthesis and implementation process (Wilson, 2015). This approach favours the optimization of the design and provides technical criteria for future extensions or adaptations of the system (Taraate, 2022).

The present work has its purpose to design and implement a synchronous digital clock by means of modular architecture on the Alhambra II FPGA platform using IceStudio as a development environment (Thakur et al., 2024), (Daza Justo & Rentero Jiménez, 2025). The proposal contemplates the integration of independent modules for the generation of the time base, the counting of seconds, minutes and hours, as well as the display interface by means of six seven-segment displays controlled through BCD decoders (Álvarez & Astudillo, 2019). Likewise, it considers the evaluation of the utilization of logic resources, the analysis of compliance with timing constraints and the experimental validation of the operation of the system on physical hardware, with the purpose of demonstrating the viability, modularity and efficiency of the developed architecture (Ramón Peula, 2020), (Vela et al., 2020).

1.1 Theoretical foundation

Within the design of digital systems, the implementation of elementary synchronous architectures constitutes a fundamental stage for understanding the operation of sequential circuits. The development of modules such as frequency dividers, counters, registers and finite state machines allows the establishment of reusable functional blocks for digital systems of greater complexity (Bonet Sanz, 2018), (Hirota & Ozawa, 2002).

In this context, the digital clock represents a widely used case study due to the integration of combinational and sequential logic, synchronization by means of a clock signal and hierarchical control of multiple interdependent modules (González Gómez & de Miguel Paraíso, 2025). The synthesis of these systems on FPGA platforms requires the use of hardware description languages (VHDL, Verilog) and design tools that allow the behavioural or structural description of the system to be translated into a configuration of bits that programmed the logic resources of the device (Mano et al., 2005). Among the theoretical concepts that support the development of a synchronous digital clock are (Garza Garza, 1998), (Rubio, 2008), (Torabi, 2011):

1. Synchronous architectures: Digital systems whose operation is governed by a common clock signal that synchronizes all the transition edges of the memory elements (flip-flops), guaranteeing the stability and predictability of the temporal behaviour.
2. Frequency dividers: Modules that, from a high-frequency clock signal (typically that of the oscillator of the FPGA board), generate lower-frequency signals by means of modulo-N counters, thus establishing the time base for the counting of seconds.
3. Counters and registers: Sequential elements that store and update states as a function of the clock signal. BCD (Binary-Coded Decimal) counters allow decimal digits (0–9) to be represented in binary form for subsequent display.
4. Finite state machines (FSM): Computational models that describe the behaviour of a sequential system by means of states, transitions and conditions, used in the hierarchical control of the clock modules (for example, reset, time setting, format change).
5. BCD to seven-segment decoders: Combinational circuits that convert a 4-bit BCD value into the signals necessary to activate the corresponding segments of a seven-segment display, allowing the visualization of decimal digits.
6. Timing constraints: Specifications that define the time requirements that the synthesized design must meet, such as setup times, hold times and maximum operating frequency, to guarantee that the system functions correctly on the physical hardware.
7. Synthesis and implementation: Process by means of which the HDL description is translated into a network of gates and memory elements optimized for the specific resources of the FPGA, followed by mapping, placement and routing, which determines the physical assignment of the modules in the matrix of logic cells of the device.

The understanding and application of these theoretical fundamentals are indispensable for addressing the modular design, the functional verification and the experimental validation of digital systems such as the digital clock proposed in the present work. The primary contribution of the present work consists in establishing and validating a complete, traceable design flow (from FSM specification to physical FPGA implementation) applicable as a reproducible methodological framework for synchronous sequential digital systems. The remainder of this paper is organized as follows: Section 2 describes the hardware platform and development environment; Section 3 presents the general system architecture; Section 4 details the logical design based on state machines, excitation tables and Karnaugh map minimization; Section 5 addresses the physical implementation and synthesis process; Section 6 reports the experimental validation; and Section 7 discusses the results and identifies directions for future work.

2 Hardware and Platform Description

The implementation of the proposed digital system requires the utilization of a development platform based on FPGA technology, with sufficient capacity to integrate combinational and sequential logic within a single programmable device.

The selection of a platform with these characteristics allows the construction of completely customized digital architectures, as well as the experimental validation of designs developed using hardware synthesis tools.

For the present work, the Alhambra II FPGA Board development board is employed, a platform oriented both towards learning and towards the development of embedded digital systems. Its architecture incorporates an FPGA device from the Lattice iCE40HX4K family, widely recognized for presenting an adequate relationship between logic capacity, energy consumption and ease of implementation in low and medium complexity applications. The reconfigurable nature of the device allows the internal architecture to be completely modified by loading a new configuration file, without requiring physical modifications to the development board (León et al., 2018).

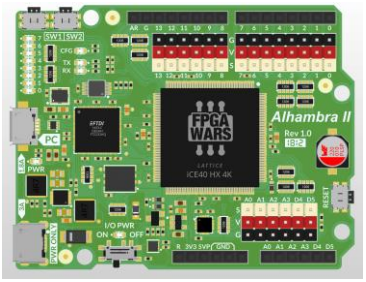

Fig. 1. Top view of the Alhambra II development board (Rev. 1.0) featuring the Lattice iCE40HX4K FPGA, FTDI FT2232HQ USB interface, general-purpose I/O headers, and on-board power regulation circuitry.

The Lattice iCE40HX4K FPGA integrates approximately four thousand logic cells, distributed in programmable blocks capable of implementing combinational and sequential functions. Likewise, it incorporates dedicated blocks for distributed memory, programmable routing resources, global clock distribution networks and specialized elements for the implementation of registers, counters and finite state machines. The availability of these resources allows the development of digital systems of considerable complexity while maintaining a moderate occupation of the device. From the point of view of the physical interface, the Alhambra II board has multiple general-purpose input and output pins (GPIO), intended for the connection of external devices such as switches, push-buttons, sensors, communication modules and display systems. In the development of the present project, these pins are used to establish communication between the FPGA and six seven-segment display modules, each controlled by means of a 74LS48 BCD to seven-segment decoder integrated circuit.

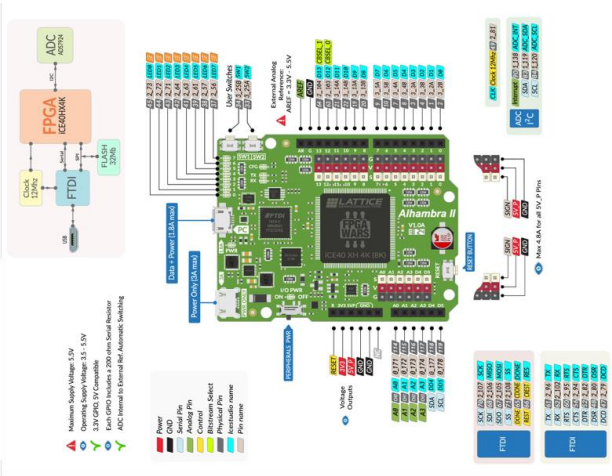

Fig. 2. GPIO pin assignment diagram of the Alhambra II V1.0A development board.

This configuration allows the control logic implemented within the FPGA to be decoupled from the circuit responsible for exciting each of the luminous segments. The generation of the visual information requires the implementation of six independent four-bit buses in BCD format, corresponding to the units and tens of seconds, minutes and hours. Subsequently, each BCD code is transmitted to its respective 74LS48 decoder, responsible for automatically converting the binary information into the signals necessary to activate the segments of the display. This strategy allows the internal architecture of the FPGA to be simplified by delegating the decoding function to the external integrated circuit, in addition to favoring a modular structure during the design of the system.

The temporal synchronization of the digital clock depends on the main clock signal available on the Alhambra II board. Due to the high frequency of the incorporated oscillator, it is necessary to implement a frequency divider module capable of

generating a time base of 1 Hz, essential for guaranteeing the correct increment of the seconds, minutes and hours counters. The use of a single clock domain allows the completely synchronous behaviour of the system to be maintained, reduces race conditions and minimises problems associated with propagation delays between modules.

The development of the logic circuit is carried out using the IceStudio graphical environment, an open-source tool designed to facilitate the programming of FPGA devices belonging to the Lattice iCE40 family. Unlike conventional development based exclusively on hardware description languages, IceStudio provides a design methodology by means of interconnected functional blocks, which allows combinational modules, registers, counters, frequency dividers and other digital components to be represented graphically. This methodology favours the understanding of the system architecture, increases the reusability of modules and simplifies the debugging process during the different stages of development. The design flow begins with the construction of the individual modules corresponding to the frequency divider, seconds, minutes and hours counters, as well as the blocks responsible for managing the reset and propagation conditions between each stage of the digital clock. Subsequently, the hierarchical integration of all the modules is carried out until the complete architecture of the system is formed.

The process culminates with logic synthesis, physical pin assignment, generation of the configuration file (bitstream) and programming of the FPGA installed on the Alhambra II board. The combined use of the Alhambra II platform and the IceStudio environment provides a suitable development environment for validating synchronous digital architectures implemented entirely in reconfigurable hardware. Likewise, it facilitates the subsequent analysis of aspects related to the utilisation of logic resources, the compliance with timing constraints and the efficiency of the developed modular architecture, fundamental elements for the technical evaluation of the proposed system.

3 Design and Implementation

The architecture of the system is structured by means of a hierarchical and modular organization, with the purpose of facilitating the development, integration, functional validation and reusability of each of the digital blocks that make up the clock. A design methodology based on independent modules is adopted, which allows the analysis of the logical behaviour to be simplified, the complexity of the global system to be reduced and future modifications or extensions of the architecture to be favoured without affecting the operation of the other components.

The general structure of the system comprises five main functional blocks: a frequency divider, the counting modules corresponding to seconds, minutes and hours, the control logic for synchronization between stages and the subsystem responsible for the generation of the information destined for the display devices. Each block is developed independently and subsequently integrated within the FPGA by means of hierarchical connections established in the IceStudio environment.

The operation of the system begins with the frequency divider block. Due to the high frequency provided by the main oscillator of the Alhambra II board, it is necessary to carry out a controlled reduction of the clock signal until obtaining a frequency equivalent to one pulse per second (1 Hz). To this end, a binary counter is implemented capable of dividing the original frequency by means of successive synchronous stages, generating a stable time base and providing the synchronization signal for the rest of the architecture. The use of a single master clock allows the completely synchronous behaviour of the system to be maintained and avoids problems associated with timing differences between modules.

Once the one-second signal is available, the time counting is carried out by means of a hierarchical structure of synchronous counters. The first block corresponds to the seconds counter, consisting of two independent stages intended for the storage of the units and tens of seconds. The units counter allows the representation of values between zero and nine, while the tens counter limits the counting interval between zero and five. Upon reaching the maximum permitted value (59 seconds), a carry signal is automatically generated towards the next functional block, and the seconds counting is reset without external intervention. The block corresponding to the counting of minutes implements an architecture equivalent to that used for seconds. The information is again divided into units and tens, an operating interval between 00 and 59 minutes is established and an enable pulse is propagated towards the hours counter upon completing a full counting cycle. This organization allows functional independence between modules to be maintained, the design logic to be simplified and the individual validation of each stage to be favoured before carrying out the complete integration of the system.

The block destined for the counting of hours develops a specific logic to represent the time format defined for the digital clock. The architecture considers two synchronized counters capable of storing the units and tens of hours, in addition to incorporating the necessary logic to automatically limit the operating range. The detection of the maximum permitted value activates the reset of the hour counting and allows a new time cycle to begin without altering the synchronization of the complete system.

The coordination between the different blocks is carried out by means of a control logic module responsible for managing the propagation of enable, reset and carry signals between the different counters. This block implements the logical conditions necessary to guarantee that each counter increments only when the immediately lower module completes its

operating interval. Consequently, the temporal coherence between seconds, minutes and hours is maintained throughout the entire operation of the clock. The development of this logic requires the obtaining of simplified Boolean functions by means of Karnaugh maps and Boolean algebra, allowing the number of gates implemented within the FPGA to be minimized and the utilization of logic resources to be optimized.

The visual representation of the temporal information is carried out by means of an output subsystem based on six independent buses encoded in BCD format, corresponding to the units and tens of seconds, minutes and hours. Each group of four bits transmits the information towards a 74LS48 decoder integrated circuit, responsible for converting the BCD code into the signals necessary to activate the corresponding segments of each display. This strategy allows the decoding function to be delegated to the external hardware, reducing the logic complexity implemented within the FPGA and maintaining a clear separation between the digital processing and the display stage.

The modular implementation developed in IceStudio allows each of the functional blocks to be represented graphically by means of clearly defined hierarchical connections. This methodology favours the reusability of components, simplifies the location of errors during the debugging stage and facilitates the incorporation of new functions, such as manual time setting, incorporation of alarms, timers or alternative display formats, without requiring substantial modifications to the main architecture of the system.

As a result, a completely synchronous, modular and scalable architecture is obtained, capable of integrating all the functions necessary for the operation of the digital clock within the Alhambra II FPGA. The functional independence of each block, together with the synchronization by means of a single clock domain and the hierarchical organization of the design, provide a robust structure for the implementation of sequential digital systems of greater complexity and constitute an adequate basis for the subsequent analysis of logic resource utilization and temporal behaviour of the programmable device.

4 Logical Design and Implementation

The development of the digital clock is based on a synchronous sequential logic design methodology, based on the obtaining of Boolean functions by means of Boolean algebra and Karnaugh maps, as well as on the utilisation of T-type flip-flops for the storage of the system states. This methodology allows the functional specifications of the clock to be transformed into a completely implementable architecture within the Alhambra II FPGA, by means of hierarchical modules developed in the IceStudio environment. As an initial stage of the design process, the sequential behaviour of each counter is defined by means of a finite state machine (FSM), establishing the transition sequence corresponding to the decimal system. For the units counter, the states are represented using four binary variables (Q_3, Q_2, Q_1, Q_0), obtaining the sequence;

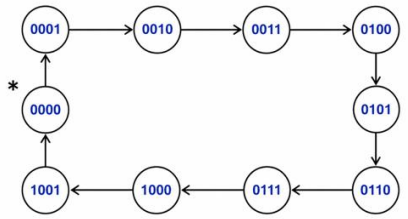

Fig. 3. State transition diagram of the synchronous BCD counter implemented with T-type flip-flops, showing the complete counting sequence from state 0000 to state 1001 with return to the initial state.

which corresponds to the operation of a synchronous decimal counter implemented by means of sequential logic. Each state represents a decimal value between zero and nine, while the transition between states depends exclusively on the excitation logic applied to the T-type flip-flops (Lee et al., 2023).

The graphical representation of the state machine allows the sequential behaviour of the counter to be visualised and the evolution of the system to be formally established. From this representation, the state transition table is elaborated, relating for each combination of the present state (Q_n) the next state (Q_{n+1}).

Table 1. State Transition and T-Type Flip-Flop Excitation Table for the Synchronous BCD Counter

Present State (Q_n)				Next State (Q_{n+1})				Inputs			
Q_3	Q_2	Q_1	Q_0	Q_3	Q_2	Q_1	Q_0	T_3	T_2	T_1	T_0
0	0	0	0	0	0	0	1	0	0	0	1
0	0	0	1	0	0	1	0	0	0	1	1
0	0	1	0	0	0	1	1	0	0	0	1
0	0	1	1	0	1	0	0	0	1	1	1
0	1	0	0	0	1	0	1	0	0	0	1
0	1	0	1	0	1	1	0	0	0	1	1
0	1	1	0	0	1	1	1	1	1	1	1
0	1	1	1	1	0	0	0	0	0	0	1
1	0	0	0	0	0	0	0	1	0	0	1

This information constitutes the basis for determining the excitation signals necessary in each T-type flip-flop. The obtaining of the excitation inputs is carried out by means of the characteristic equation of the T-type flip-flop, $T_i = Q_i \oplus Q_i^+$ where Q_i, denotes the current state of the bistable, Q_i^+ the XOR logical operation (Yan et al., 2022). This relation allows the conditions under which each flip-flop modifies its state during the active edge of the clock signal to be established, guaranteeing a completely synchronous evolution of all the system variables. Table 1 presents the complete state transition and excitation table derived from the FSM specification, from which the Boolean expressions for each T flip-flop input are subsequently obtained.

With the information obtained from the transition table, the Karnaugh maps corresponding to each excitation input (T_0, T_1, T_2 T_3), are constructed. The simplification of the Boolean functions allows the number of logic gates necessary to implement the counter to be reduced, the occupation of resources within the FPGA to be decreased and the temporal performance of the circuit to be improved during the synthesis process.

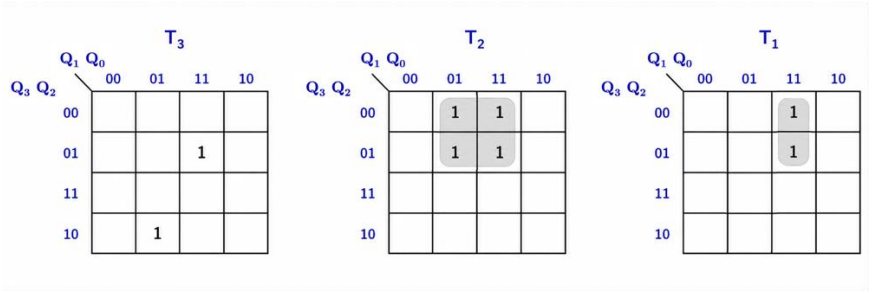

Fig. 4. Karnaugh maps for the minimisation of T-type flip-flop excitation functions T_3, T_2, and T_1 of the synchronous BCD counter, with prime implicant groupings highlighted.

The Boolean excitation functions are subsequently minimised using Karnaugh maps, as shown in Figure 4, from which the simplified expressions are extracted for implementation within the IceStudio environment. As a result of the minimisation procedure, the excitation functions are obtained:

$$\begin{aligned} T_0 &= 1 \\ T_1 &= Q_1\, Q_0\, \overline{Q}_3 \\ T_2 &= Q_0\, \overline{Q}_3 \\ T_3 &= Q_1\, Q_0\, \overline{Q}_3\, Q_2 + \overline{Q}_1\, Q_0\, \overline{Q}_3\, Q_2 \end{aligned} \qquad (1)$$

which correspond to the Boolean expressions physically implemented within the FPGA. Each equation defines the necessary conditions to produce the switching of the associated flip-flop, allowing the decimal sequence required by the synchronous counter to be generated. The equation corresponding to T_0 establishes the permanent switching of the least significant bit, while the remaining functions incorporate progressively more complex logical conditions to guarantee the correct transition between states. The sequential behaviour of the system is described by means of the characteristic equation of the T-type flip-flop, $Q_i(k+1) = Q_i(k) \oplus T_i(k)$, where $Q_i(k)$ represents the present state of the bistable, $Q_i(k+1)$ and $T_i(k)$, the corresponding excitation logic function. This formulation allows the temporal evolution of the counter to be modelled mathematically, and the simultaneous updating mechanism implemented within the FPGA to be described.

The implementation of the Boolean equations is carried out by means of AND, OR and NOT logic gates automatically synthesised by IceStudio during the compilation process. Each excitation function is connected directly to the corresponding input of the T-type flip-flops, obtaining a completely synchronous counter where all state variables are updated simultaneously on the active edge of the clock signal. This strategy allows the cumulative delays characteristic of asynchronous counters to be eliminated, the temporal stability of the system to be increased and the correct synchronisation between all the implemented modules to be guaranteed.

The decimal counter developed constitutes the basic functional block for the construction of the complete digital clock. The reutilisation of this architecture allows the counters corresponding to the units of seconds and minutes to be implemented, while for the tens additional limiting logic is incorporated with the purpose of restricting the counting interval between zero and five. In an analogous manner, the module corresponding to the counting of hours requires specific Boolean functions to automatically limit the operation to the defined time format, maintaining the global synchronisation of the system at all times.

The communication between the different modules is carried out by means of carry signals, generated when a counter completes its operation cycle. These signals act as enable conditions for the immediately higher counter, allowing the hierarchical propagation of the count between seconds, minutes and hours. The utilisation of this strategy favours the functional independence between modules, simplifies the integration process and facilitates the reutilisation of blocks during the development of the complete system.

The general synchronisation of the clock is based on a single clock signal coming from the oscillator of the Alhambra II board. Due to the high frequency available in the device, a frequency divider is previously implemented, responsible for generating a one hertz signal, used as a temporal reference for all the counters of the system. The utilisation of a single clock domain allows race conditions to be avoided, problems associated with propagation delays to be reduced, and the simultaneous updating of all the registers implemented within the FPGA to be guaranteed.

Finally, the information generated by each counter is transmitted through six independent buses encoded in BCD format towards the 74LS48 integrated circuits. Each decoder automatically converts the BCD code into the signals necessary for the driving of the seven-segment displays, allowing the units and tens of seconds, minutes and hours to be visually represented. This architecture allows the processing logic implemented within the FPGA to be decoupled from the display stage, a modular organisation of the design to be maintained and future system extensions to be facilitated.

As a result of the described procedure, a completely synchronous, modular and optimised architecture is obtained, based on the synthesis of finite state machines, Boolean functions minimised by means of Karnaugh maps and sequential storage using T-type flip-flops. The implementation developed on the Alhambra II FPGA allows the proposed logic design to be experimentally validated, demonstrating the correct integration between the theoretical modelling, the hardware synthesis and the physical implementation of the digital clock.

5 Physical Implementation and FPGA Programming

Once the logic design of the digital clock has been concluded and the individual functionality of each of the modules has been verified, the physical implementation of the system is carried out by means of the FPGA incorporated in the Alhambra II board. This stage comprises the complete process of logic synthesis, physical resource allocation, generation of the configuration file and programming of the reconfigurable device, allowing the architecture developed in the design environment to be transferred towards a functional implementation in hardware.

5.1 Synthesis

The implementation flow begins with the structural description of the system within the IceStudio environment, hierarchically integrating the modules corresponding to the frequency divider, seconds, minutes and hours counters, control logic and output buses intended for display. The modular organisation of the project allows functional independence between each block to be maintained, the debugging process to be simplified and the reusability of components to be favoured during the integration of the complete system. The IceStudio graphical implementation of the seconds counter subsystem, comprising the units and tens BCD counters with their respective T flip-flop excitation and carry logic, is shown in Figure 5.

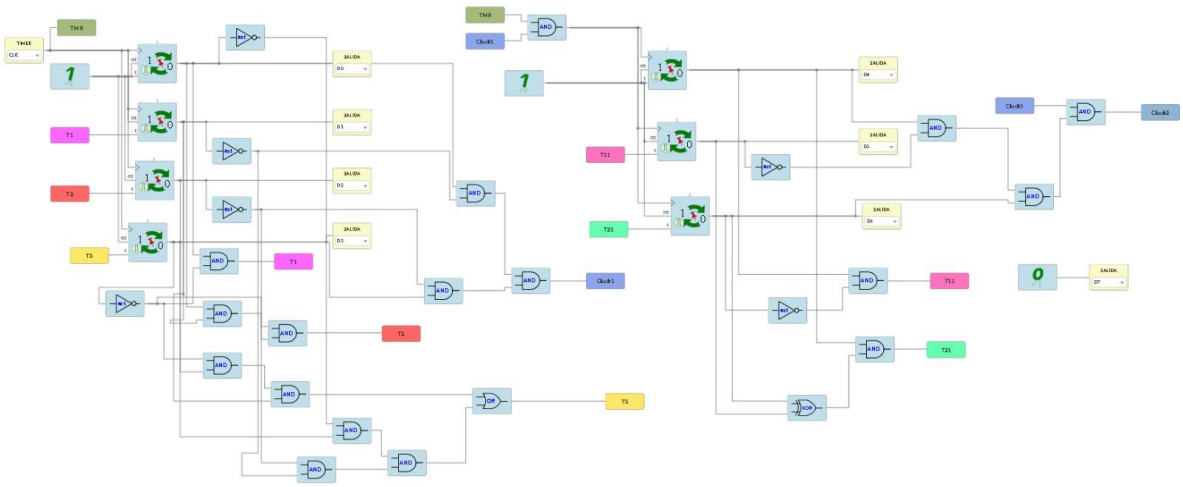

Fig. 5. IceStudio block diagram implementation of the seconds counter subsystem: units-of-seconds BCD counter (left) and tens-of-seconds BCD counter (right), including T flip-flop excitation logic, combinational carry generation (Clock1, Clock2), and BCD output lines (D0–D7).

Subsequently, the logic synthesis process is executed, the objective of which consists in transforming the functional description of the circuit into an equivalent network of logic elements compatible with the internal architecture of the FPGA. During this stage, the Boolean equations obtained by means of Karnaugh maps, the connections between T-type flip-flops and the associated combinational logic are interpreted, automatically carrying out the optimisation of the implemented functions. As a result, a technological representation composed of logic cells, registers, interconnections and resources specific to the programmable device is generated. The synthesis also allows the structural consistency of the design to be verified, possible connection conflicts to be detected, unused signals to be identified and the correct definition of the inputs, outputs and internal registers to be validated. The absence of errors during this stage constitutes an indispensable requirement for continuing with the physical implementation process within the device.

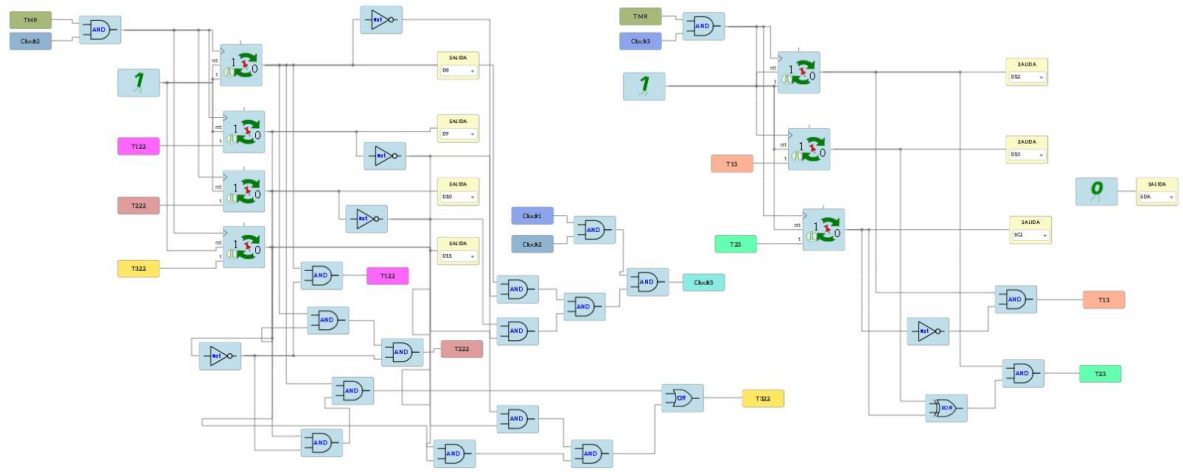

Fig. 6. IceStudio block diagram implementation of the minutes and hours counter subsystem: tens-of-minutes BCD counter (left), carry propagation logic generating Clock3 (centre), and units-of-hours counter (right), with BCD output lines D8–D12 and seven-segment display interface signals SDA and SC1.

The implementation of the minutes and hours subsystems, together with the hierarchical carry propagation logic that interconnects all counter stages, is presented in Figure 6.

5.2 Place & Route

Once the synthesis is completed, the Place & Route stage is executed, corresponding to the physical allocation of the internal resources of the FPGA. During this procedure, the location of each logic block within the programmable matrix is automatically determined and the interconnection routes necessary to guarantee communication between the different modules of the system are established. The optimisation of the routing process allows propagation delays to be minimised,

the length of the interconnections to be reduced and compliance with the timing constraints established by the development tool to be favoured.

The Placement algorithm assigns each flip-flop, logic gate and functional block to a specific position within the FPGA, considering the availability of resources and the proximity between related elements. Subsequently, the Routing process determines the physical connections by means of the programmable interconnection network available in the device, guaranteeing electrical continuity between all the implemented modules and preserving the behaviour defined during the logic design stage.

5.3 Generation of the bitstream and configuration in the FPGA

Once the physical implementation process has been concluded, the configuration file (bitstream) is automatically generated, which contains the necessary information to completely define the internal architecture of the FPGA. This file stores the configuration corresponding to the Boolean functions, registers, interconnection routes, pin assignment and synchronization resources implemented during the synthesis and routing process. The programming of the device is carried out by means of the USB interface incorporated in the Alhambra II board, using the tools integrated in IceStudio to transfer the configuration file towards the programming memory of the FPGA. During this stage, the programmable logic matrix is completely configured, enabling the operation of all the previously designed modules and allowing the execution of the digital clock directly on the reconfigurable hardware.

5.4 Pin assignment and functional verification

The physical pin assignment is carried out by means of the explicit definition of each input and output signal of the system, establishing the correspondence between the ports defined in the logic design and the connectors available on the development board. Particularly, the BCD outputs corresponding to units and tens of seconds, minutes and hours are associated with the 74LS48 integrated circuits, responsible for carrying out the conversion towards the control signals of the seven-segment displays. Likewise, the main clock signal is assigned to the oscillator incorporated in the board, guaranteeing a single temporal reference for all the implemented modules.

Subsequently, the functional verification of the system is carried out by means of experimental tests on physical implementation. The correct generation of the one-second signal by the frequency divider is checked, the counting sequence corresponding to seconds, minutes and hours is verified, as well as the adequate propagation of the carry signals between the different modules. Finally, the visual representation of the time is validated by means of the six seven-segment displays, confirming the correct decoding of the BCD codes provided by the FPGA towards the 74LS48 integrated circuits.

5.5 Experimental validation

The implementation of the system on the Alhambra II FPGA allows the correspondence between the logic model developed by means of state machines, Boolean equations and Karnaugh maps, and the behaviour observed in the programmable hardware, to be experimentally checked. Likewise, it demonstrates the viability of a design methodology based on modular architecture, optimized logic synthesis and synchronization by means of a single clock domain, favouring the scalability of the system and allowing future extensions without modifying the fundamental structure of the design.

6 Validation and Experimental Results

The validation of the digital clock is carried out by means of the functional simulation of the logic design developed in IceStudio and the experimental verification of its operation on the Alhambra II board. This procedure allows the correspondence between the theoretical model obtained from finite state machines, Boolean equations simplified by means of Karnaugh maps and the behaviour observed during the physical implementation of the system to be verified.

6.1 Functional simulation

As a first stage, the functional simulation of each of the modules that make up the architecture of the digital clock is carried out. The evaluation comprises the frequency divider, the synchronous counters implemented by means of T-type flip-flops, the logic for generating the excitation signals (T_0, T_1 T_2 T_3), the carry propagation circuits and the outputs intended for time display. The simulation allows the correct transition sequence between states to be checked and the independent operation of each block to be verified before carrying out the integration of the complete system.

Subsequently, all the modules are integrated within a single hierarchical architecture using IceStudio. The structural implementation allows the interconnection between the frequency divider, the counters corresponding to seconds, minutes and hours, the combinational logic responsible for generating the excitation functions and the output signals intended for the display system to be established. The modular organization of the design facilitates the integration, debugging and validation process of the complete circuit.

The synthesis of the design generates a completely functional logic architecture, composed of AND, OR, NOT and XOR gates, as well as T-type flip-flops synchronized by means of a single clock domain. The implementation obtained preserves the structure proposed during the logic design stage, evidencing the correspondence between the Boolean equations derived from the Karnaugh maps and the circuit synthesized within the FPGA.

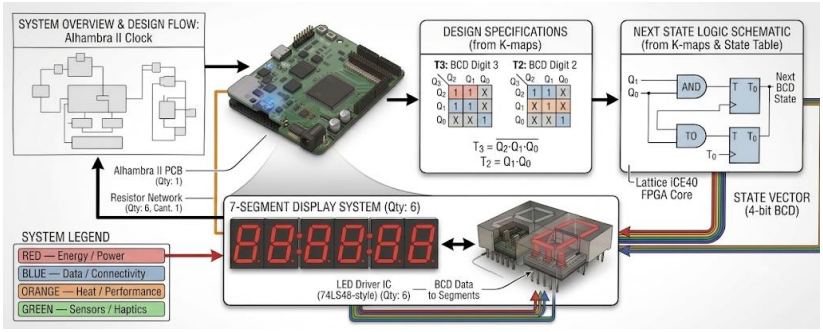

Fig. 7. Conceptual system overview and design flow of the Alhambra II-based synchronous digital clock: FSM-derived excitation logic for the BCD counter (T_2, T_3), next-state logic schematic implemented on the Lattice iCE40HX4K core, and six-digit seven-segment display subsystem driven by 74LS48-style BCD decoders.

Figure 7 presents a conceptual overview of the complete system design flow, illustrating the correspondence between the FSM-based logical design, the excitation logic synthesized on the Lattice iCE40HX4K core, and the seven-segment display output subsystem.

6.2 Programming and experimental configuration

Once the implementation process has been concluded, the programming of the Alhambra II FPGA is carried out by means of the configuration file generated by IceStudio. The experimental validation is performed by connecting the outputs of the FPGA to six 74LS48 integrated circuits, responsible for converting the BCD codes into the signals necessary for the driving of the seven-segment displays. This configuration allows the units and tens corresponding to seconds, minutes and hours to be visualised in real time.

During the experimental tests, the continuous operation of the digital clock is checked by means of the direct observation of the counting sequence. The correct increment of the units and tens of seconds is verified, the automatic generation of the carry signals upon completing each counting cycle and the corresponding updating of the minutes and hours counters. Likewise, the adequate operation of the reset logic implemented to prevent the appearance of invalid states during transitions between consecutive cycles is confirmed.

6.3 Experimental results

The precision of the system depends on the stability of the oscillator integrated in the Alhambra II board and on the correct operation of the frequency divider implemented within the logic design. The utilisation of a single clock signal allows uniform synchronisation to be maintained between all the sequential modules, favouring the simultaneous updating of the registers and avoiding inconsistencies during the counting process. During the tests carried out, stable behaviour of the system is observed, without presenting synchronisation errors or appreciable alterations in the temporal sequence.

The physical implementation confirms the correct integration between the combinational logic responsible for generating the excitation functions of the T-type flip-flops and the sequential logic responsible for the storage of the system states. Likewise, the correspondence between the model developed by means of finite state machines, the Boolean equations obtained by simplification with Karnaugh maps and the behaviour observed on the programmable platform is verified.

6.4 Validation of the complete system

As a result, the operation of the digital clock implemented on the Alhambra II FPGA is experimentally validated, demonstrating the viability of a design methodology based on modular architecture, synchronisation by means of a single clock domain and logic synthesis based on fundamentals of Boolean algebra and sequential logic. The developed implementation allows it to be verified that the procedure followed during the design produces a functional, stable and adequate system for digital electronics applications based on programmable logic devices.

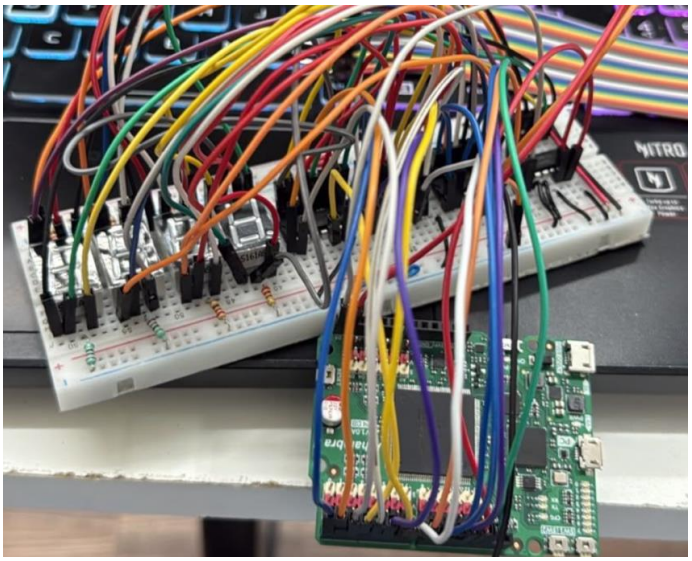

Fig. 8. Physical prototype implementation.

In the initial experimental configuration shown in the figure, the circuit was developed physically. Additionally, a contrast exists between the EDA mapping tool and the real application, since this is a prototype where the assembly is different and the cables are not optimized. Within the test environment, the setup is completed using the Alhambra II FPGA board to empirically verify the logical correctness of the synchronous digital clock and therefore the viability of the FPGA core under real, non-ideal laboratory conditions, successfully confirming that the state transitions and the 7-segment display drivers operate correctly before proceeding with final hardware optimization.

7 Discussion

The proposed architecture allows the viability of implementing a fully functional digital clock by means of a design methodology based on sequential logic to be evidenced, avoiding the utilization of predefined blocks or proprietary modules available in various FPGA development tools. Unlike implementations based on integrated counters, embedded processors or high-level descriptions, the developed system is derived directly from the obtaining of Boolean equations by means of Boolean algebra and Karnaugh maps, allowing an explicit relationship to be established between the mathematical model, the logic synthesis and the physical implementation of the circuit.

7.1 Methodological contribution

The present work is not limited to the implementation of a digital clock as an ultimate end but presents it as a case study to validate a complete and reproducible logic design methodology. In this sense, the true contribution of the work consists in demonstrating a systematic development sequence:

$$\text{Specification} \rightarrow \text{FSM} \rightarrow \text{State Table} \rightarrow \text{Excitation Table} \rightarrow \text{Karnaugh Maps} \rightarrow \text{Boolean Equations} \rightarrow \text{IceStudio} \rightarrow \text{FPGA} \quad (2)$$

This methodological flow proves reproducible for other sequential systems of diverse nature and constitutes a more solid academic contribution than the isolated implementation of a single digital clock. From the perspective of a scientific publication, this approach provides greater technical value and strengthens the originality of the work, by transferring the emphasis from the final product towards the design procedure and its applicability in broader contexts. In comparison with other designs reported for similar applications, the developed methodology presents a high degree of traceability during all stages of the design. The progressive construction of the system from finite state machines, transition tables, excitation tables for T-type flip-flops and minimised Boolean functions facilitates the understanding of the internal operation of the circuit and provides a reproducible methodological structure for the development of sequential digital systems of greater complexity. This characteristic represents a significant advantage within the academic field, where the logical analysis of the system constitutes an objective as important as the final implementation.

7.2 Advantages of modular architecture

Another relevant advantage corresponds to the modular organisation of the design. The functional separation between the frequency divider, the seconds, minutes and hours counters, the control logic and the display stage allows each block to be developed, verified and integrated independently. This strategy simplifies the debugging process, favours the reusability of modules and facilitates future extensions without modifying the general architecture of the system. Likewise, the functional

independence between blocks contributes to maintaining an organised and scalable structure during the synthesis and implementation process. From the perspective of logic resource utilisation, the developed architecture bases its operation exclusively on AND, OR, NOT and XOR gates, as well as on T-type flip-flops synchronised by means of a single clock domain. The absence of embedded processors, dedicated memories or specialised blocks allows an implementation with reduced logic complexity and an efficient utilisation of the resources available in the FPGA to be inferred. This characteristic favours the incorporation of new functions without significantly compromising the capacity of the programmable device. The synchronisation strategy employed constitutes another relevant aspect of the design. The distribution of a single clock signal to all the sequential elements allows the cumulative delays characteristic of asynchronous architectures to be eliminated and the simultaneous updating of all the registers to be guaranteed. As a consequence, stable temporal behaviour is obtained during the experimental validation and a correct propagation of the carry signals between the different counters implemented.

7.3 Limitations of the system

Nevertheless, the developed system presents some limitations inherent to the methodology employed. The temporal reference depends exclusively on the oscillator integrated in the Alhambra II board and on the frequency, divider implemented by means of digital logic. Consequently, any accumulated deviation associated with the frequency of the oscillator directly affects the absolute precision of the clock during prolonged periods of operation. Likewise, the absence of permanent storage mechanisms prevents the temporal information from being preserved after interrupting the electrical power supply of the system. Another limitation corresponds to the absence of communication interfaces with external devices, restricting the operation of the clock to a completely autonomous operation. The implemented architecture also does not incorporate mechanisms for automatic time setting, synchronisation with external temporal references or additional functions intended to increase interaction with the user.

7.4 Future work

As future work, it is advisable to incorporate a dedicated Real Time Clock (RTC) module, allowing the temporal information to be maintained even in the absence of main power by means of an independent backup source. The integration of this device would significantly increase the precision and reliability of the system during continuous operation applications. In a complementary manner, the implementation of external synchronisation mechanisms by means of digital communication protocols is considered, enabling the automatic updating of the temporal information from external devices or higher precision reference systems. This characteristic would extend the field of application of design towards distributed systems, automation platforms and interconnected embedded architectures. Finally, the incorporation of additional functions, such as programmable alarms, configurable timers, digital stopwatches, electronic calendars or serial communication interfaces, would allow the proposed architecture to be transformed into a platform of greater functional capacity without modifying the fundamental logic structure developed during this work. Thanks to the modular nature of the design, such extensions would be carried out by means of the integration of new functional blocks, preserving the hierarchical organisation and the design methodology implemented on the Alhambra II FPGA.

For future implementations, a system is developed where the design is applied to a printed circuit board. Therefore, the design includes adding synchronization through an RTC clock, which allows the PCB to be smaller in size. It also includes a display for showing the time zone, along with reduced wiring, which leads to less interference and fewer connection errors. Additionally, it incorporates the necessary electronics such as resistors, a crystal oscillator, power supplies, and output pins for configuring the clock with the external environment.

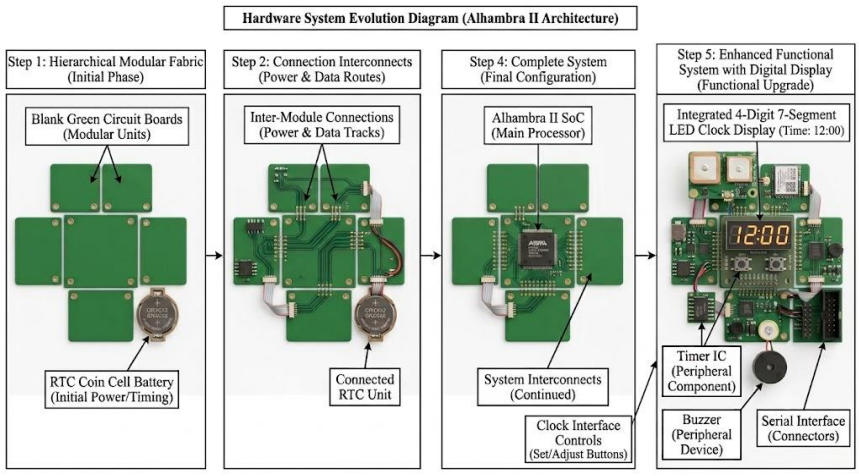

Fig. 9. Evolution of the Architecture for Future Implementations.

8 Conclusion

The present work demonstrates the feasibility of a complete, traceable, and reproducible design methodology for synchronous sequential digital systems, applied to the implementation of a modular digital clock on the Alhambra II FPGA using IceStudio as the open-source development environment. Beyond the functional artefact itself, the central contribution of this work resides in establishing and validating a systematic design flow that progresses (without gaps or auxiliary tools) from the formal specification of a digital problem to its experimental verification on physical programmable hardware. This flow constitutes a methodological framework directly applicable to a broad class of sequential digital systems and provides a reproducible reference for academic and embedded engineering contexts alike.

The design flow developed follows a well-defined sequence: functional specification through finite state machines (FSMs), elaboration of state transition tables, derivation of excitation inputs for T-type flip-flops via the characteristic equation $T_i = Q_i \oplus Q_i^+$, minimisation of the resulting Boolean expressions through Karnaugh maps, and synthesis and implementation of the optimised logic within the IceStudio graphical environment. The consecutive and rigorous application of these stages transforms a high-level design problem into a completely synchronous, modular, and experimentally verified architecture, preserving at every stage a direct and transparent correspondence between the mathematical model and the behaviour of the hardware. This traceability (from FSM to FPGA) constitutes the distinguishing characteristic of the methodology and represents its primary contribution to the fields of digital electronics education and embedded systems engineering.

From a formal design standpoint, the minimisation of Boolean excitation functions via Karnaugh maps proved to be one of the most impactful stages of the methodology. The simplification of the expressions $T_0 = 1$, $T_1 = Q_1Q_0Q_3'$, $T_2 = Q_0Q_3'$, and $T_3 = Q_1Q_0Q_3'Q_2 + Q_1'Q_0Q_3Q_2$ directly reduced the combinational logic required within the FPGA, decreasing the number of logic cells occupied and improving the timing performance of the synthesised design. The use of T-type flip-flops, synchronised by a single global clock domain derived from the on-board oscillator via a binary frequency divider, ensures the simultaneous updating of all state variables on each active clock edge. This strategy effectively eliminates the race conditions and cumulative propagation delays that characterise asynchronous architectures, resulting in a temporally stable and functionally consistent sequential system throughout the duration of the experimental validation.

The modular organisation of the architecture constitutes a further structural contribution of the present work. The decomposition of the system into functionally independent blocks (a frequency divider, synchronous BCD counters for units and tens of seconds, minutes, and hours, a hierarchical carry propagation and control logic module, and a six-channel BCD output subsystem interfaced to 74LS48 decoders and seven-segment displays) allows each component to be designed, verified, and integrated individually without affecting the behaviour of the remaining modules. This functional independence simplifies the debugging process, reduces the complexity of the global integration stage, and directly favours the reusability of each block in future designs of greater scope. The hierarchical organisation also provides a clear structural template for the development of more complex sequential systems, such as digital calendars, programmable timers, event counters, or multi-format timing devices, all of which can be built upon the same architectural foundation without requiring a fundamental restructuring of the core design.

The experimental validation carried out on the Alhambra II FPGA confirms the correct operation of the complete system under real hardware conditions. The frequency divider generates a stable 1 Hz time base, the BCD counters for seconds, minutes, and hours advance correctly through their respective counting sequences, carry signals propagate accurately between stages, and the six seven-segment displays continuously represent the temporal information without synchronisation errors or invalid state occurrences. The correspondence between the logical model derived from the FSM-based design procedure and the behaviour observed during physical execution on the programmable device provides direct experimental evidence of the internal consistency of the proposed methodology. This alignment between formal design and hardware outcome is not incidental; it is the result of the structured and stepwise nature of the design flow, which eliminates ambiguities at each stage before proceeding to the next.

The use of the Alhambra II platform and the IceStudio graphical synthesis environment deserves specific consideration within the conclusions. The open-source nature of both tools removes the economic and licensing barriers that typically restrict access to FPGA-based development in resource-constrained educational settings. The block-based, graphical design paradigm offered by IceStudio maintains a direct visual correspondence between the structural description of the circuit and its logical organisation, allowing students and early-career engineers to engage with FPGA design without requiring prior mastery of hardware description languages such as VHDL or Verilog. This accessibility, combined with the Lattice iCE40HX4K device's adequate logic capacity for educational and mid-complexity applications, positions the Alhambra II ecosystem as a technically sound and pedagogically appropriate platform for the type of work presented here. The successful implementation of a complete, multi-module synchronous architecture on this platform reinforces its suitability for digital electronics instruction and for low-cost embedded system development.

It is equally important to acknowledge the limitations inherent to the current implementation. The temporal reference of the clock depends exclusively on the stability of the Alhambra II on-board oscillator and on the binary frequency divider implemented in digital logic. Any accumulated frequency deviation of the oscillator translates directly into a drift in the

absolute precision of the clock during extended periods of operation. Furthermore, the absence of non-volatile storage prevents the system from preserving the current time value after a power interruption, and the lack of an external synchronisation interface restricts the clock to fully autonomous operation without the possibility of correction by an external temporal reference. These limitations do not undermine the validity of the design methodology, but they define the boundaries of the current implementation and motivate the future developments described below.

Several directions for future work are identified as natural extensions of the present contributions. The integration of a dedicated Real Time Clock (RTC) module, operating with an independent battery-backed oscillator, would provide significantly higher temporal precision and ensure the preservation of the time value across power interruptions, extending the practical applicability of the system to continuous operation contexts. The incorporation of serial communication interfaces (including I²C, SPI, or UART) would enable external synchronisation with higher-precision time references, networked time servers, or host systems, transforming the current autonomous implementation into a node within a broader distributed or embedded architecture. Additionally, the modular nature of the developed architecture directly facilitates the incremental addition of new functional blocks, including programmable alarms, countdown timers, electronic stopwatches, digital calendars, and multi-format display modes, without requiring modifications to the fundamental logic structure already implemented and validated. Each of these extensions can be incorporated as independent hierarchical modules within the existing IceStudio project, preserving the design methodology and the single-clock-domain synchronisation strategy that constitute the core of the present work.

At a broader level, the present work provides evidence that the implementation of synchronous digital systems on FPGA platforms does not require dependence on automated high-level synthesis tools, embedded processor cores, or proprietary intellectual property blocks. A rigorous command of the foundational principles of sequential logic (state machine specification, excitation table analysis, Boolean minimisation, and synchronous design) is sufficient to produce functional, stable, and resource-efficient implementations on modern reconfigurable hardware. The design flow validated here is, by construction, platform-agnostic in its formal stages and requires only minimal adaptation for deployment on alternative FPGA families or development environments. This generality constitutes a methodological contribution of direct relevance to educational technology, embedded systems engineering, and the broader community of practitioners engaged with open-source programmable logic development.

In summary, the present work establishes a validated, complete, and reproducible design flow for synchronous FPGA architectures, demonstrated through the implementation of a modular digital clock on the open-source Alhambra II platform. The methodology integrates classical digital design formalism with practical synthesis tools, produces an experimentally verified hardware implementation, and provides a scalable technical foundation for the development of more complex sequential digital systems. The results confirm the coherence between the theoretical model, the logic synthesis, and the physical implementation, consolidating the proposed approach as a solid methodological basis for future reconfigurable digital applications within the fields of electronic engineering, embedded systems, and engineering education.

References

Álvarez, L., & Astudillo, C. (2019). Diseño y desarrollo de entrenador de FPGA libre de bajo costo basado en FPGA Alhambra II. *Revista Juventud y Ciencia Solidaria*, (2).

Ashenden, P. J. (2007). *Digital design (Verilog): An embedded systems approach using Verilog*. Elsevier.

Bolaños, M. R., Boette, A. P., Videla, F. A., & Rebón, L. (2023). Desarrollo de un sistema para el conteo de múltiples señales en coincidencia basado en tecnología FPGA. *Anales AFA, 34*(1), 27–32. https://doi.org/10.31527/analesafa.2023.34.1.27

Bonet Sanz, M. (2018). *Desarrollo de una arquitectura y un conjunto de reglas para diseño de FPGAs desde lenguaje C en entornos "Do It Yourself"* [Trabajo Fin de Grado, Universidad Politécnica de Madrid].

Club Robótica Granada. (2019, March 10). *Introducción a las FPGA's libres*. https://clubroboticagranada.github.io/taller-fpgas-libres/

Cortez, A., Franco, Z., & Borjas, J. (2023). Noise generator by free FPGA technology. *Universidad, Ciencia y Tecnología, 27*(121), 64–73. https://doi.org/10.47460/uct.v27i121.755

Daza Justo, J., & Rentero Jiménez, J. R. (2025). *Sistemas empotrados sobre Alhambra-II* [Trabajo de Fin de Grado, Universidad Complutense de Madrid].

Dillien, P. (2017, March 6). *And the winner of best FPGA of 2016 is…* EE Times.

FPGAwars. (n.d.). *Alhambra II FPGA board* [Hardware repository]. GitHub. Retrieved September 7, 2026, from https://github.com/FPGAwars/Alhambra-II-FPGA

Freire Rubio, M. Á. (2008). *Diseño síncrono de circuitos digitales* [Material docente]. Universidad Politécnica de Madrid.

Garza Garza, J. Á. (1998). *Diseño lógico programable* [Master's thesis, Universidad Autónoma de Nuevo León].

González Gómez, J. (n.d.). *FPGA wars*. Retrieved September 7, 2026, from http://obijuan.github.io/intro-fpga.html

González Gómez, J., & de Miguel Paraíso, M. Á. (2025). *Arquitectura de computadores: Laboratorio*. Universidad Rey Juan Carlos.

Hirota, K., & Ozawa, K. (1989). The concept of fuzzy flip-flop. *IEEE Transactions on Systems, Man, and Cybernetics, 19*(5), 980–997. https://doi.org/10.1109/21.44013

Juárez León, F., Ramírez Agundis, A., & Díaz Carmona, J. (2014). PBL, un caso de estudio: Diseño de un reloj digital con base en un FPGA y un monitor VGA. *Pistas Educativas, 35*(108), 1757–1771.

Kilts, S. (2007). *Advanced FPGA design: Architecture, implementation, and optimization*. Wiley-IEEE Press. https://doi.org/10.1002/9780470127896

Lee, G., Joo, B., & Kong, B. S. (2023). CMOS clock-gated synchronous up/down counter with high-speed local clock generation and compact toggle flip-flop. *IEEE Transactions on Circuits and Systems I: Regular Papers, 70*(12), 5316–5327. https://doi.org/10.1109/TCSI.2023.3307611

Leong, P. H. W. (2008). Recent trends in FPGA architectures and applications. In *2008 4th IEEE International Symposium on Electronic Design, Test and Applications (DELTA 2008)* (pp. 137–141). IEEE. https://doi.org/10.1109/DELTA.2008.14

López Monterroso, J. M. (2020). *FPGA libre Alhambra II como herramienta de aprendizaje para la electrónica digital* [Trabajo de graduación, Universidad de San Carlos de Guatemala].

Mano, M. M., & Kime, C. R. (2005). *Fundamentos de diseño lógico y de computadoras* (3.ª ed.). Pearson Educación.

Navarro-Torrero, P., Rojas-Muñoz, L. F., Martínez-Rodríguez, M. C., Barriga-Barros, Á., Jiménez-Fernández, C. J., Brox, M., & Brox, P. (2024). Digital design flow based on open tools for programmable logic devices. In *2024 XVI Congreso de Tecnología, Aprendizaje y Enseñanza de la Electrónica (TAEE)* (pp. 1–5). IEEE. https://doi.org/10.1109/TAEE59541.2024.10604893

Ramón Peula, M. (2020). *Desenvolupament d'un curs d'aprenentatge en línia d'introducció a l'electrònica digital mitjançant l'ús de FPGAs lliures* [Treball Final de Grau, Universitat de Lleida].

Saiz-Vela, A., Fontova, P., Pallejà, T., Tresanchez, M., Garriga, J. A., & Roig, C. (2020). Plataforma de desarrollo de bajo coste para implementar circuitos digitales en FPGAs mediante hardware y software libre. In *XIV Congreso de Tecnologías Aplicadas a la Enseñanza de la Electrónica: Actas TAEE 2020* (pp. 419–428). Instituto Superior de Engenharia do Porto.

Simpson, P. (2010). *FPGA design: Best practices for team-based design*. Springer. https://doi.org/10.1007/978-1-4419-6339-0

Taraate, V. (2022). *Digital logic design using Verilog: Coding and RTL synthesis* (2nd ed.). Springer. https://doi.org/10.1007/978-981-16-3199-3

Thakur, S., Ahmad, B., Pearce, H., Tan, B., Dolan-Gavitt, B., Karri, R., & Garg, S. (2024). VeriGen: A large language model for Verilog code generation. *ACM Transactions on Design Automation of Electronic Systems, 29*(3), 1–31. https://doi.org/10.1145/3643681

Torabi, M. (2011). A new architecture for T flip flop using quantum-dot cellular automata. In *2011 3rd Asia Symposium on Quality Electronic Design (ASQED)* (pp. 296–300). IEEE. https://doi.org/10.1109/ASQED.2011.6111764

Wilson, P. (2015). *Design recipes for FPGAs: Using Verilog and VHDL* (2nd ed.). Newnes. https://doi.org/10.1016/C2010-0-65668-9

Wolf, W. (2004). *FPGA-based system design*. Prentice Hall PTR.

Yan, A., Liu, R., Huang, Z., Girard, P., & Wen, X. (2022). Designs of level-sensitive T flip-flops and polar encoders based on two XOR/XNOR gates. *Electronics, 11*(10), 1658. https://doi.org/10.3390/electronics11101658