Sequential circuits often use flip-flops (FFs) or latches for data storage. Latches have advantages in error-resilient applications, lower supply voltage operation, reduced power consumption, and increased operating frequency. However, complex timing constraints have limited their adoption in commercial and industrial designs. To overcome this limitation, researchers have explored the automatic conversion of FF-based designs into latch-based designs, primarily focusing on performance enhancement by reducing the clock period and considering potential area improvements.

Different solutions have been proposed, including pulsed latch designs, multiphase clocking schemes and retiming methodologies. All of them have specific drawbacks that limit their deployment in industrial design flows which consist in: preventing pulse signal degradation in all operating conditions, increasing the area due to additional retimed registers, lack of fomal verification methodology or requiring multiple clocks generation and complex clock distribution networks.

In this thesis, we introduce a methodology called *Mix&Latch*, designed to address the mentioned limitations.

The key-points of the proposed flow are: trasforming flip-flop designs into positive transparent latches (PTLs) based designs that leverage time borrowing, incorporating negative transparent latches (NTLs) as retention barriers, instead of relying on delay padding, to address short-path hold constraints, employing a single clock tree throughout the design and merging adjacent latch pairs into positive-edge-triggered flops (PETFs) or negative-edge-triggered flops (NETFs) to reduce area overhead.

The first part of this thesis provides a comprehensive explanation of the proposed methodology. It describes the modeling of circuit timing and positional data, the conversion of the optimization problem into an integer linear programming (ILP) form and the enhancements made to the original methodology.

The second part of this thesis presents the experimental results obtained. In the initial version, experimental evaluations demonstrate the advantages of this approach on a suite of benchmark circuits. The enhancements to the algorithm are then evaluated on a *RISC-V* processor, showing a reduction in the implementation flow runtime, diminished area overhead, and enhanced timing performance in comparison to retiming, which was executed using a state-of-the-art commercial tool.