Asynchronous Dataflow Synthesis onto FPGAs using the eTeak Framework

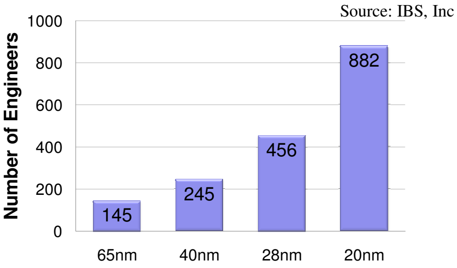
Mahdi Jelodari Mamaghani, Jim Garside and Steve Furber
The University of Manchester
School of Computer Science
Contact: eTeak@cs.man.ac.uk

1. Moore's Legacy Everywhere

Hardware System Design requires hardware engineers to learn Hardware Description Languages (HDLs) to implement a diverse set of applications from dedicated processing units to CPUs , GPUs, NPUs, etc.

Aggressive technology scaling demands more hardware engineers per project due to the ever increasing complexity of Hardware systems which raises challenges in clock distribution, power budget management, etc.

Source: IBS, Inc



4. eTeak: A Synchronous Elastic Dataflow Synthesiser

eTeak aims to bridge the gap between software and hardware domains. In other words, eTeak enables software programmers to exploit the emergent capabilities of hardware systems. eTeak raises the design abstraction to algorithmic level to provide the designer with flexibility in implementation of concurrent hardware regardless of timing. This approach conveys potential advantages in the context of GALS systems.

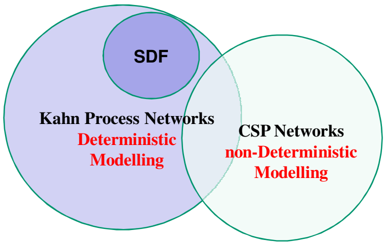
eTeak features can be grouped into communication and computation facets: eTeak networks are synthesised in a syntax-directed compilation manner from a CSP-like language. The primitives of the language, including channels and processes, are preserved which form point-to-point communication between the computation blocks at hardware level which contributes to concurrent message passing.

- **Slack Elasticity** allows a system to be pipelined with any degree of storage (e.g. Arbitrary size FIFOs) on its communication channels. This behaviour was first formalised for the distributed computation systems. Slack Elasticity provides a flexible communication environment for the computational blocks in the system. eTeak take advantage of this property to optimise the processes without affecting the overall functionality of the system. Composition and decomposition of modules towards GALSifcation benefits from elastic communication which is not possible in the synchronous domain where rigid timing is adopted since the initial design stage.
- **The Macromodule logic** enables implementing complex circuits using simple data processing building blocks. This concept simplifies the asynchronous control design. eTeak employs this technique to perform the control interactions locally instead of having them as a separate central unit which has significant performance implications. eTeak exploit the local (aka distributed) control behaviour to apply "De-elastisation" within macromodules which results in defining new boundaries in the network.
- **Point-to-Point communication** enables a module to have independent rates of data streaming from different sources which contributes to a higher level of concurrency and accordingly effective throughput. This model allows development of parallel computations both in software and hardware domains. In an industrial context it has been leveraged for scaling computations, meanwhile it has been employed in academia to implement low power computation platforms.
- **The Memory Architecture**: The `push' nature of the dataflow systems pose a significant design barrier when it comes to memories and storage elements. In conventional designs memory holds data tokens; processing unit keep reading them when required through `pull' channels. In dataflow architectures memories are handled in a different fashion such that after every read the associated memory location needs to be re-written to avoid data loss in the next cycle. eTeak has a hybrid model: it leverages push channels for computation and pull channels for memory handling.

2. Channel-based Programming

Aggressive technology up scaling has raised the following issues for hardware designers and demands further engineering work to tackle the emergent complexities:

- ❑ Mismatch between gate and interconnect delays
- ❑ Variability in terms of power and clock speed
- ❑ Difficulties in power budget management
- ❑ Difficulties in clock distribution within a chip



Meanwhile in the software domain, engineers are exploiting channel based computing paradigm to overcome the scaling issues.

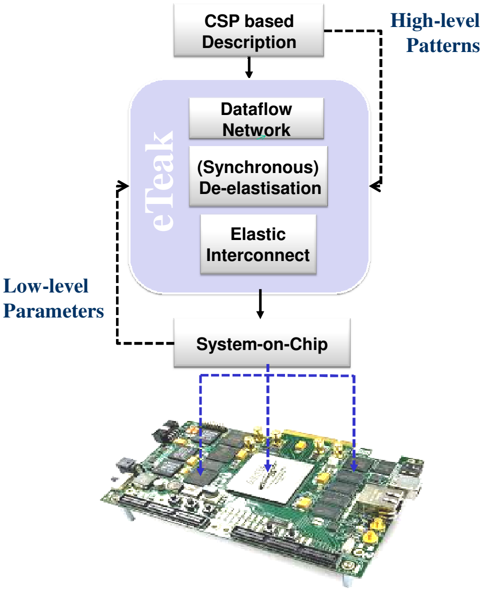
5. Automatic 'Clock'

The Clock is the most important signal in a hardware system as it regulates the execution rate and orchestrates the computation.

Therefore, it is not a good idea to reflect it to the software programmers!

The eTeak synthesis system introduces a *global timing discipline* to the circuit, then attempts to optimise the circuit based on *High-level* and *Low-level* parameters: This technique is named *De-elastisation* [DATE'15].

Verilog Netlist = eTeak(High-level Patterns, low-level Parameters)

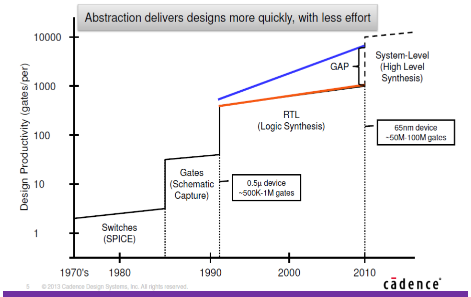


3. The software vs. hardware gap!

The ever increasing complexity of System-on-Chips~(SoCs) calls for new design technologies to enhance designer productivity and bridge the software/hardware gap. The High-level Synthesis technology abstracts hardware design at a higher level, describing dataflows rather than Register Transfer Level~(RTL) interactions.

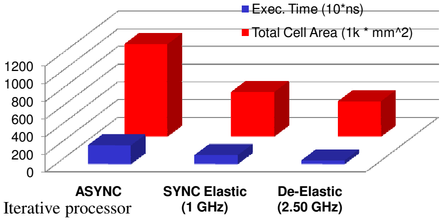
Much like a software description, these high-level flows neglect implementation details such as clocks and are, naturally, asynchronous and elastic. This abstraction allows the designer to focus on specifying the system functionality and postpone issues of timing to subsequent stages of the synthesis flow.

But how efficient are these high-level flows?

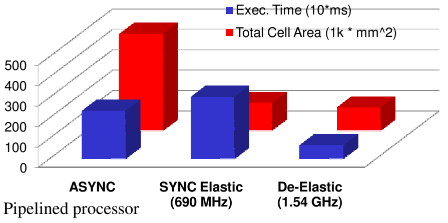


6. How Efficient is it?

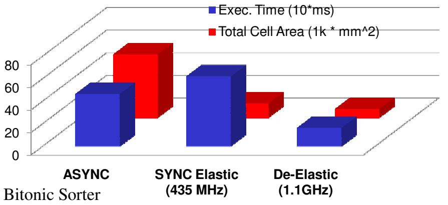
To compare the software and hardware level concurrency to demonstrate the software realisation of the same architecture running on a laptop and let the audience compare the hardware vs. software concurrency. We have also compared a set of synchronous and asynchronous designs in terms of throughput and area. As is it is shown below De-elastic eTeak outperforms the asynchronous designs by a factor of 3-4.



Iterative processor



Pipelined processor



Bitonic Sorter

7. Future Work

We demonstrate eTeak as a novel approach for system-level design. It incorporates synchronous subcircuits in an asynchronous elastic flow to achieve higher performance while minimizing the area. Automatic partitioning the system into structural loops with different critical paths opens up a new synthesis scheme for High-level GALS synthesis. As future work, we will focus on running "de-elastisized" parts with different clock frequencies within elastic ecosystem, such as mutli-FPGA systems with heterogeneity..

This work was supported by EPSRC Grant **"Globally Asynchronous Elastic Logic Synthesis (GAELS)"** (EP/I038306/1), and a full research scholarship from the School of Computer Science, The University of Manchester.