OpenTransputer: Reinventing a parallel machine from the past

Student: David Keller, Andres Amaya Garcia, Supervisor: Prof. David May, Project Type: Enterprise

University of Bristol, Department of Computer Science

Introduction

- In the 1980s a company called Inmos developed a revolutionary machine called the 'Transputer' together with the parallel programming language Occam.
- The Transputer is a device intended for parallel computing. It features on-chip memory and serial communication links. Unfortunately, it was released in a time when parallelism was not the main concern, so it did not receive the attention it deserved.
- Nowadays, there is increasing interest in making computing more ubiquitous. The size and ease with which microprocessors communicate with other devices are paramount in modern applications such as the Internet of Things. We consider that this is an environment in which the Transputer could outperform other architectures in the market.

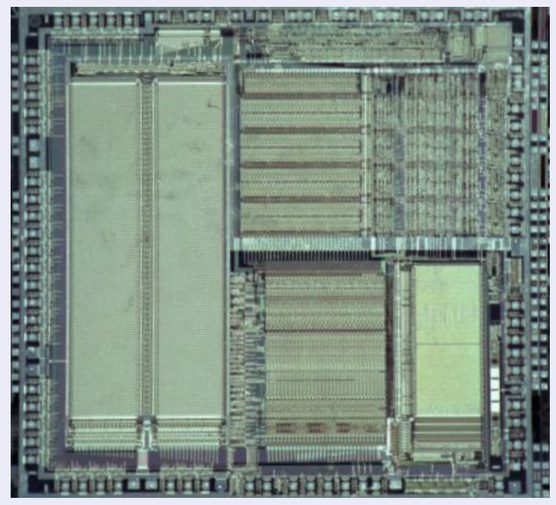

Figure 1: Inmos Transputer T414

1. Objectives

- Our aim is to re-implement the microarchitecture of the original Transputer Instruction Set Architecture (ISA). The intention is to modernise the design to make it more attractive for solving problems in 2015.
- The new design maintains all the original concepts and ideas of the original device for concurrency management and interprocess communication. However, the serial links that were used to connect multiple Transputers are replaced by a switch that routes messages between cores.
- We intend to introduce a new I/O interface that is compatible with the vast majority of hardware components such as accelerometers and gyroscopes. This will make the machine very attractive for applications involving wearable computing and the Internet of Things. We expect the new interface to be easy to use while delivering optimal performance.

2. Business Plan/Research Proposal

- The OpenTransputer is a tool for the emergent Internet of Things and wearables markets.
- Simple yet powerful building block processor with unique set of features an attractive entry in the multi-billion dollar embedded chip market.
- Our portfolio: open source processor design, consulting services (integration into products etc.), licensing of add-ons such as floating-point or vector units.

open hardware

3. Processor internals

Some of the processor instructions are complex and can take many clock cycles to complete. Thus, we have decided to break down each instruction into multiple (very simple) microinstructions that can be completed in one cycle. The processor can be viewed as a large Finite State Machine (FSM) where each microinstruction is a state. Therefore, each assembly instruction is no more than a pointer to a start state. Microinstructions have an associated set of signals that control the processor datapath.

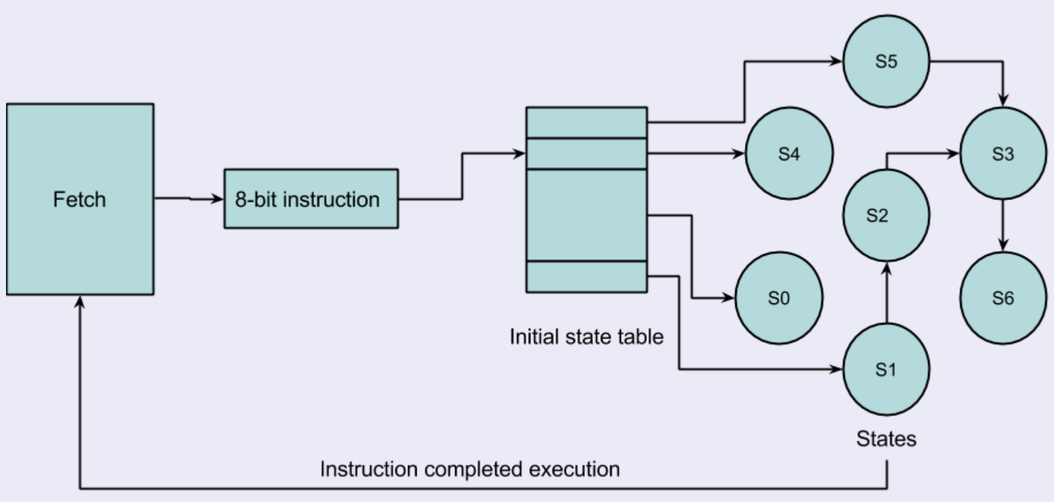

Figure 2: State machine view of the processor microarchitecture

4. Network

- Switches form a Beneš network, allowing for rearrangeably non-blocking communication.
- An unused input can always be connected to an unused output in this kind of network.
- Each switch has 2 inputs and 2 outputs.
- In total there are r switches and $N = 2 \cdot r$ inputs and outputs.

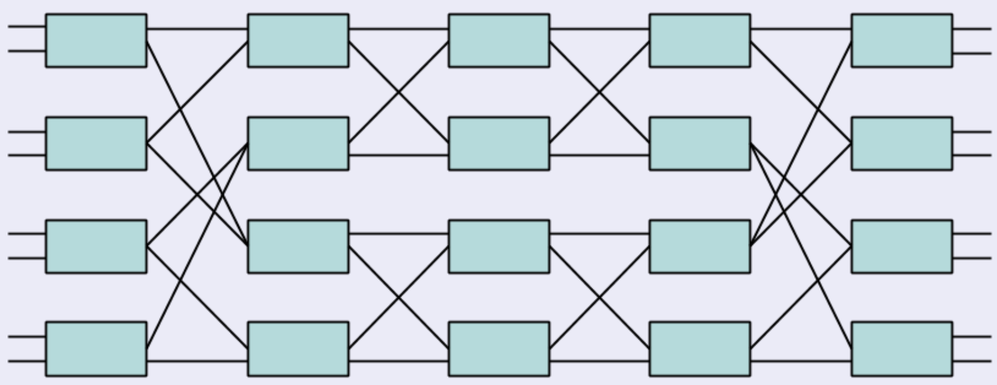

Figure 3: Layout of Beneš network

5. Progress and Status

1. Implemented in Verilog HDL the core components of the processor. This includes basic arithmetic, memory access and scheduling instructions.
2. Developed and integrated a switch that enables us to connect multiple processors into a network.
3. We will shortly start development of the I/O interface. Extensive research must be conducted to ensure the final product offers broad compatibility with state-of-the-art hardware components.

University of BRISTOL