The Transputer

The CSP processor

Introduction

- Transputer is a processor introduced in the 1980s
- Focuses on concurrency and multiprocessing
  - Programmed using Occam
  - Features a hardware scheduler
  - Features priority queues
  - Features communication with other Transputers
- It came in three different configurations
  - T2xx 16 bit processors
  - T4xx 32 bit processors
  - T8xx 32 bit processors with an FPU
- Lost the battle against 8-bit processors

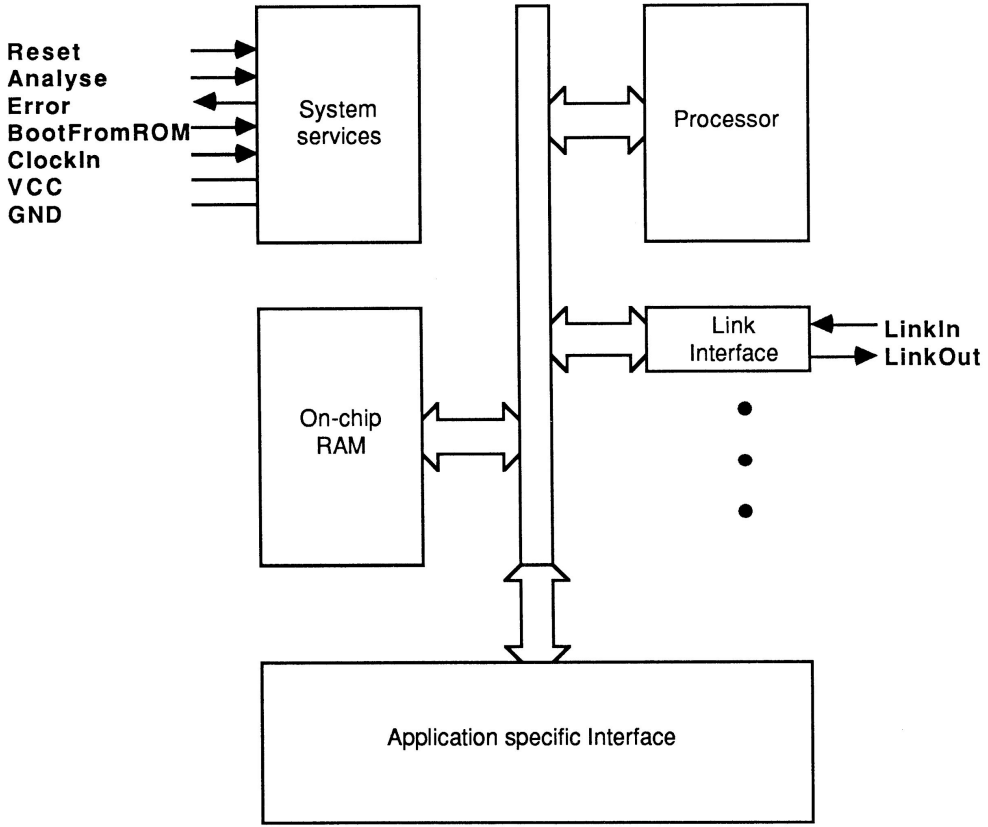
Occam

- Occam is an CSP programming language
  - Processes
  - Channels
- Occam was created in the 1980s
- Was originally made as the native language for the Transputer

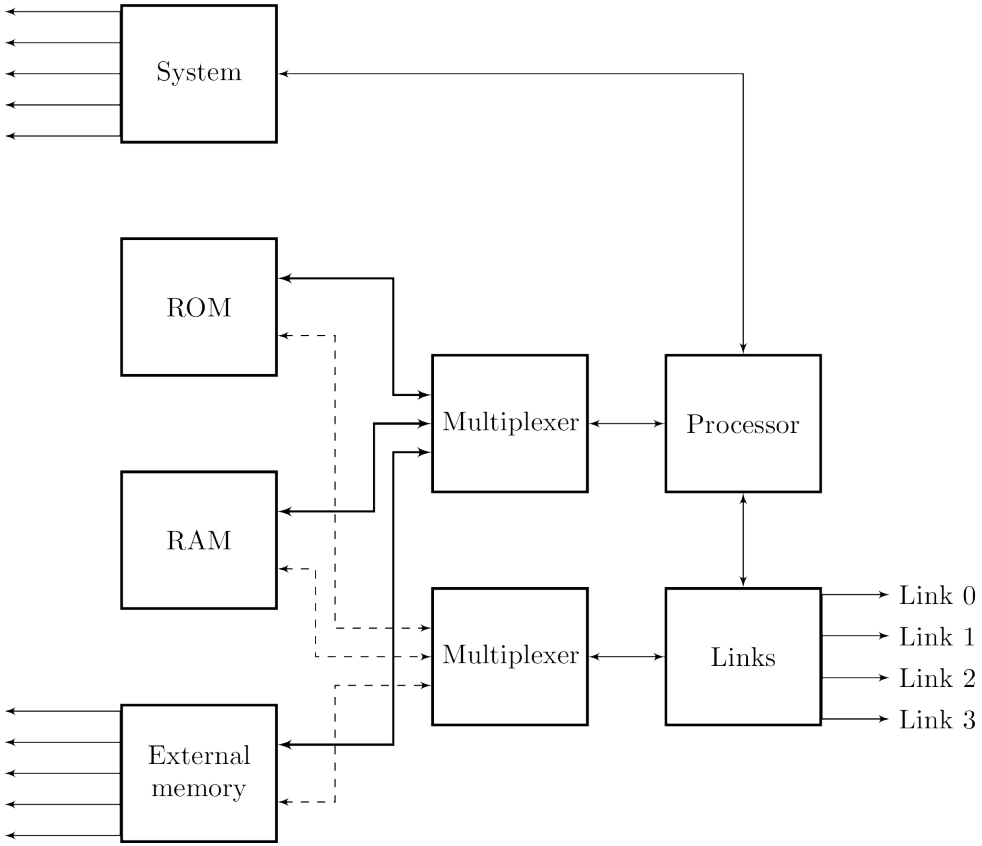
Occam - string printer

```
PROC printer (CHAN OF BYTE in, CHAN OF SP keyboard, screen)
    #USE "hostio.lib"
    INT i:
    BYTE tmp:
    SEQ
        i := 0
        WHILE i < 10:
            SEQ
                in ? tmp
                so.write.char(keyboard, screen, tmp)
                i := i + 1
:

PROC generator (CHAN OF BYTE out)
    INT i:
    BYTE gen:
    SEQ
        gen := 48 -- ascii for '0'
        i := 0
        WHILE i < 10:
            SEQ
                out ! gen
                gen := gen + 1
                i := i + 1
:

PROC main (CHAN OF SP keyboard, screen, []INT buffer)
    CHAN OF BYTE gen_to_print:
    PAR
        Generator(gen_to_print)
        Printer(gen_to_print, keyboard, screen)
:
```

Block diagram - theoretical



Block diagram - actual



Transputer concurrency

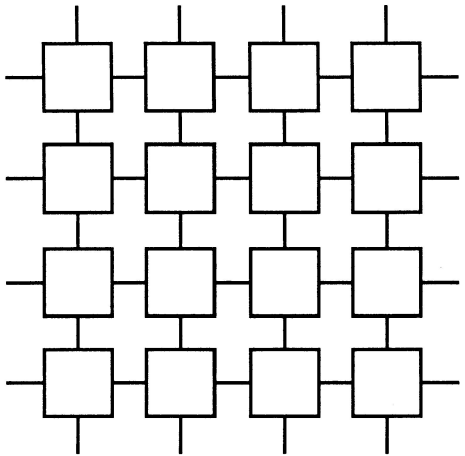
The Transputer was made for running without an operating system

In order to run multiple processes - it features an hardware scheduler

Furthermore, it was intended to communicate with several Transputers

The scheduler contains two queues: high and low priority

Each queue is a linked list of running processes



Internal Transputer

- 4 general purpose registers: a, b, c and an operand register
- Context is the four registers, instruction pointer and workspace pointer
- Memory is mainly intended as stack based
- The instructions of the Transputer are 8 bit
  - 4 bit opcode
  - 4 bit operand
  - Original argument: only 80 % instructions are commonly used
- However, longer instructions can be “build” in the operand register
- Communication can both be amongst internal processes using memory, or external Transputers using the links