Author	Dylan Muir
Student no.	01915932
Address	96 Birdwood Tce. Auchenflower, QLD 4066

Author	Dylan Muir
Supervisor	Joaquin Sitte
Project Title	A Replacement Transputer Node for the Agora Distributed Computing Architecture for Robotics
Year	2001

This project report was submitted as part of the requirements for the award of the

Bachelor of Engineering (Electronic) /
Bachelor of Information Technology (Computing Science)

in the

School of Electrical and Electronic Systems Engineering
Faculty of Built Environment and Engineering

and the

School of Computing Science and Software Engineering
Faculty of Information Technology

with cooperation from the

Smart Devices Laboratory

at the

Queensland University of Technology
Brisbane, Australia

Signature of Candidate ______________________________

A Replacement Transputer Node for the Agora Distributed Computing Architecture for Robotics

Abstract

The QUTy mini-robot is a small autonomous robotic architecture designed for research [Malmstrom, 2001]. It differs from most mini-robots in its highly modular nature. Individual hardware modules connect to each other via a simple serial bus interface based on the Transputer serial link architecture [Inmos, 1992]. Modules can be added and removed from the resulting system without requiring software or hardware modifications.

The Transputer is designed to be the core of a distributed computing architecture. Each node in a Transputer network has some processing power and a small amount of local memory. All communication across the network takes place over dedicated point-to-point serial links between nodes. Each Transputer has the ability to execute concurrent processes. The serial Transputer links do not suffer from common problems associated with standard parallel monolithic busses. Bandwidth on a serial channel is not shared between multiple peripherals, so the overall communications bandwidth is kept high. The simple serial interface provides hardware de-coupling of components, making the design of a modular system relatively easy.

The Agora architecture is a modular hardware design utilised by the QUTy mini-robot, and is based on a Transputer network. The architecture specifies the hardware separation of robot peripheral modules. Robot components are encapsulated on separate peripheral boards, which stack together in a modular fashion. A conceptual "hardware" process running on each peripheral board communicates with a control process running on a Transputer processor node.

The Transputer is no longer being manufactured by Inmos. This report outlines the development of a replacement processor node for the Agora architecture. The system processor is replaced by a modern Hitachi SH4 microprocessor. The serial links of the Agora architecture are emulated by custom logic implemented in an FPGA (Field-Programmable Gate Array). In this project an evaluation board was produced to test the feasibility of such a design. This board has been used to control the QUTy mini-robot via an adapter board to interface to the Agora architecture.

The software implementation of the serial link controller in VHDL is described in a separate report of the same name [Hobson, 2001].

Table of Contents

Table of Figures

Glossary

Agora architecture: A distributed computing architecture designed for use in small robotics applications. Based on the Transputer architecture, and used in the QUTy mini-robot.

AMBA AHB bus: An **A**dvanced **H**igh-performance **B**us developed by ARM, Ltd. Used for fast internal busses inside microprocessors.

Bitstream: The data required to configure an FPGA. Known as a bitstream due to the serial nature of the configuration upload.

BGA: Ball Grid Array. A method of mounting integrated circuit packages to a PCB. An array of solder balls is melted to the bottom of the chip, and bonded to the surface of a PCB. An example is shown in Figure 10.

Boundary scan: See JTAG.

Buffer: A device capable of strongly driving a signal, where another component may not be able to drive the signal with sufficient strength.

CLB: Configurable Logic Block. The basic re-configurable component of an FPGA.

CMOS: Complimentary Metal-Oxide Semiconductor. A technology for manufacturing integrated circuits.

Communicating process: The basic process element in the Transputer architecture. Processes communicate with each other only via the Transputer serial links.

Crystal resonator: A component used to generate a high-speed clock signal.

DRAM; Dynamic RAM. A type of memory with row/column addressing. Requires periodic refreshing via external circuitry.

DIP: Dual Inline Package. A method of mounting integrated circuit packages to a PCB. A row of pins is arranged on either side of the package. An example is shown in Figure 14.

DLL: Delay-Locked Loop. The digital equivalent of a Phase-Locked Loop (PLL). Used to account for clock skew caused by distribution delay in an FPGA routing matrix.

Endian: The organisation of words in memory. Big-endian organisation places the most significant byte lowest in memory, while little-endian organisation places the least significant byte lowest in memory.

FPGA: Field-Programmable Gate Array. A volatile programmable-logic device capable of implementing high-speed asynchronous logic designs.

Ground plane: A signal layer connected to ground, sometimes added to either side of a PCB to reduce RF emissions.

I/O: Input / Output. Usually refers to a data signal.

JTAG: An IEEE standard (1149.1) interface, also known as boundary scan. Permits the sampling of all external signal pins of a compliant chip. Also used by the Xilinx components for configuration upload.

LED: Light-Emitting Diode.

Monolithic bus: A standard parallel microprocessor peripheral bus. A single monolithic bus is used as the communications channel in most microprocessor systems.

MPX: A memory access interface supported by the SH4. Used for connecting to external memory controllers. Uses a multiplexed address and data bus.

PCB: Printed Circuit Board.

PCI: Personal Computer Interface. An extensible monolithic bus architecture used in desktop computers.

PROM: Programmable ROM. A memory array that contains user-defined data. The Xilinx range of PROMS are re-programmable in-system.

QFP: Quad Flat Pack. A method of mounting integrated circuit packages to a PCB. Sets of tinned surface-mount leads are arranged around the edges of the package. An example is shown in Figure 11.

QUTy: An autonomous mini-robot designed by [Malmstrom, 2001].

RISC: Reduced Instruction Set Computer. RISC processors obtain fast pipelined execution speeds by limiting the machine-language instruction set.

RF: Radio-Frequency.

Routing, PCB: The process of laying out tracks between components on a PCB.

Routing matrix, FPGA: The programmable set of interconnections that combines CLBs to implement user-defined logical processes.

Serial link controller: The system component on the replacement processor node that emulates the Transputer serial links. Described in [Hobson, 2001].

SH4: A RISC microprocessor family manufactured by Hitachi.

SRAM: Static RAM. A type of memory with low interface overheads. Requires no periodic refresh, while power is applied.

Supply de-coupling: A technique to eliminate voltage supply ripple. A capacitor is attached across the voltage rails.

Surface-mount: A method of mounting components to a PCB. Component leads are mounted on the surface of the PCB, rather than requiring a hole through the board.

Transputer: A multi-tasking processor designed for use in distributed computing architectures. Incorporates a point-to-point serial communications links. Originally manufactured by Inmos Ltd.

Through-hole: A method of mounting components to a PCB. A hole is drilled through the PCB to accommodate the pins of the device.

VHDL: Very high speed integrated circuits Hardware Description Language. A relatively low-level language used to design logic for implementation in an FPGA.

Volatile: A term used to refer to non-permanent storage.

von-Neumann: A processor architecture involving a sequential fetch-execute cycle. Most modern processors are based on this architecture. Transputers, however, support concurrent processes in hardware.

Statement of Authorship

The work contained in this project report has not been previously submitted for a degree or diploma at any other tertiary educational institution. To the best of my knowledge and belief, the project report contains no material previously published or written by another person except where due reference is made.

Signed ____________________________

Date ____________________________

Acknowledgements

The author would like to thank Dr. Joaquin Sitte of the Smart Devices Laboratory for his constant support and his attitude towards bureaucracy, education and research.

Kurt Malmstrom and Inmos must be thanked separately for creating a robot architecture that is a pleasure to work with.

Thanks must also go to Melissa Penny, Clive Hobson, Ross Brown, the concept of entropy and the School of Electrical Engineering.

1 Introduction

The QUTy mini-robot is a small autonomous robotic architecture designed for research [Malmstrom, 2001]. It differs from most mini-robots in its very modular nature. Individual hardware modules connect to each other via a very simple serial bus interface based on the Transputer serial link architecture [Inmos, 1992]. Modules can be added and removed from the resulting system without requiring software or hardware modifications.

Unfortunately the processor "heart" of the architecture, the Transputer, is no longer being manufactured. To enable to continued development of the robot, the Transputer processor node must be replaced with an emulation based around a modern von-Neumann processor. The hardware design of this replacement node forms the major content of this report.

2 The Transputer Architecture

The Transputer forms the core of a distributed computing architecture [Inmos, 1992]. A group of Transputer nodes is usually arranged in a homogenous network. Each node has local processing power and a small amount of local memory. All communication between nodes in the network takes place over dedicated point-to-point serial links.

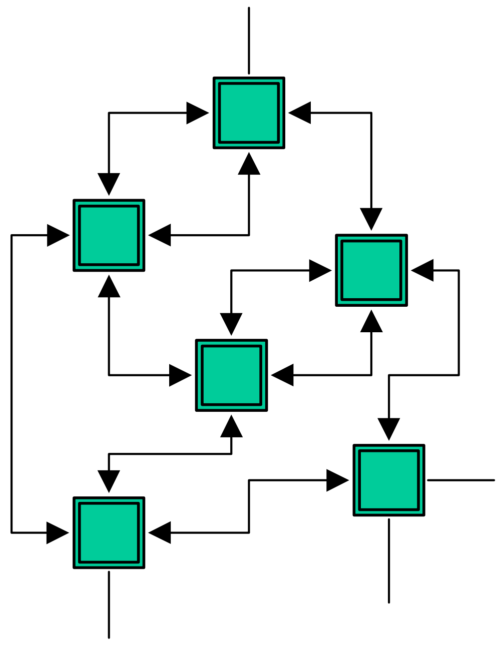


Figure 1 A Transputer newtork.

The fundamental processing elements in a Transputer network are communicating processes. Each Transputer node has the ability to execute concurrent processes. These processes send all messages via the serial link interface, whether they are communicating with a process executing on the same node or on another node in the network.

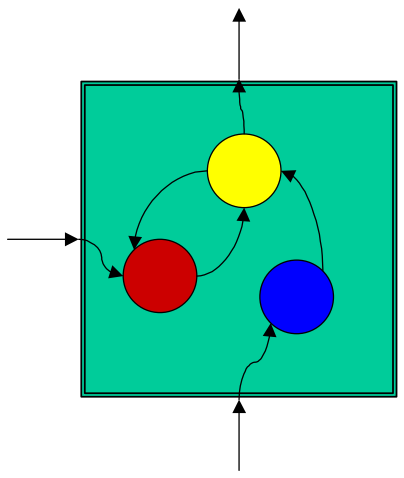


Figure 2 Concurrent processes on a Transputer node.

Figure 2 shows three concurrent processes within a single Transputer node, communicating with each other and other external processes via the serial link interface.

This consistent interface between nodes de-couples the nodes from each other. A node with a Transputer interface can transparently perform any task without requiring any special knowledge of the other nodes in the network.

These distributed communications channels are a stark contrast to the monolithic system busses used in most computing systems. Monolithic busses have a higher bandwidth than a serial link, due to their parallel nature. This comes at the cost of having a much greater number of signal lines; the Hitachi SH4 system bus can have as many as 92 signal lines running to each peripheral [Hitachi, 2000]. The bandwidth of a monolithic bus also reduces with every peripheral communicating on the bus. Transputer serial links are dedicated between nodes, and so do not suffer from this problem.

In addition to this, peripherals on a monolithic bus may need to be aware of each other to prevent communication collisions. This presents a problem when expanding the bus to incorporate more peripherals, as existing hardware may need to be modified to accommodate the additions. This makes a modular system very difficult to design.

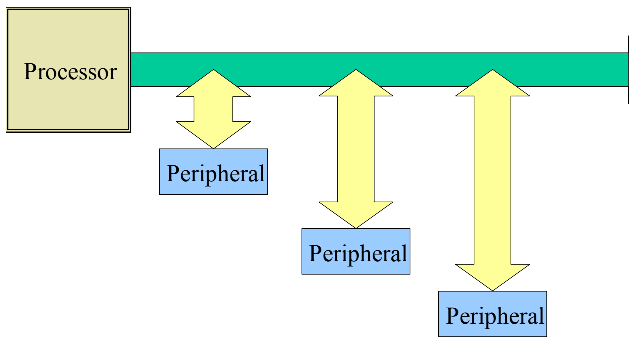


Figure 3 A von-Neumann processor with a monolithic system bus.

Modular monolithic busses exist (the PCI bus used in most desktop computers is a good example), but suffer a high overhead in both handshaking signal lines required and extra communication protocol information. A 64-bit PCI bus has 40 handshaking signal lines to support its 64-line multiplexed address and data bus.

These busses are not suited to small robotics applications, due to their physical size. The Transputer serial links require only a single signal line for a unidirectional link, or two lines for a bi-directional link. The hardware decoupling of the Transputer links makes an extensible system much easier to design. In addition, the system can be completely modular, whereas a monolithic bus specification is usually coupled to a specific processor.

2.1 The Inmos Transputer

The Inmos Transputer implementation was manufactured originally by Inmos, a company based in the United Kingdom. Inmos was bought by SGS-Thompson Microelectronics, who continued the manufacture of Transputers for some years.

The Inmos Transputer contains a micro-coded process scheduler in hardware as well as machine-code support for serial link services. The serial links themselves transmit and receive at 20 Mbits per second, which translates to a bandwidth of approximately 1.8 MBytes per second.

2.2 The Transputer Serial Links

The serial link services available to processes are provided in hardware by the Inmos Transputer. All serial link requests are synchronous, which means that the requesting process waits until the request is completed before execution continues.

The standard method for peripheral control is interrupt-driven programming. When a peripheral sends data to a process, the process receives an interrupt which it must service, irrespective of the current state of the process. The Transputer serial links enable systems-level programming without the need for interrupt handling. Link data is buffered until a control process is ready to receive it. Likewise, if no data is available a control process will halt execution until the data arrives. This simplifies program design by eliminating the requirement for re-entrant code and interrupt handlers.

The point-to-point nature of the serial links means that one separate control process is required for each peripheral. No bus multiplexing takes place, so control processes need no knowledge either of each other or the other peripherals present in the network.

The serial link protocol is very lightweight, as shown in Figure 4. No error detection or correction is incorporated, as the links are designed for low-noise environments. No routing information is required, as the links are point-to-point. No packet header is required either, as all packets are a single byte in length. Extra protocol wrapping information is left to the programmer. Two start bits and a single stop bit are used for each packet, giving a data rate of 11 bits per byte.

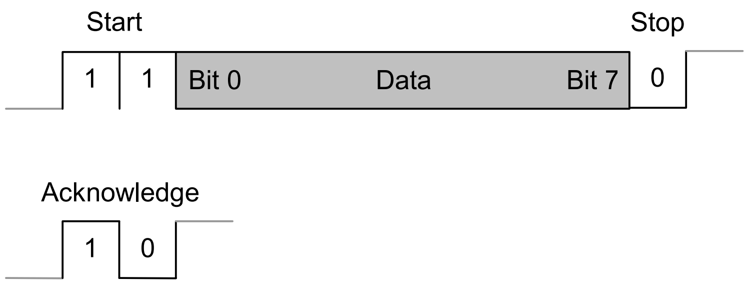


Figure 4 The Transputer serial link protocol.

3 The Agora Architecture

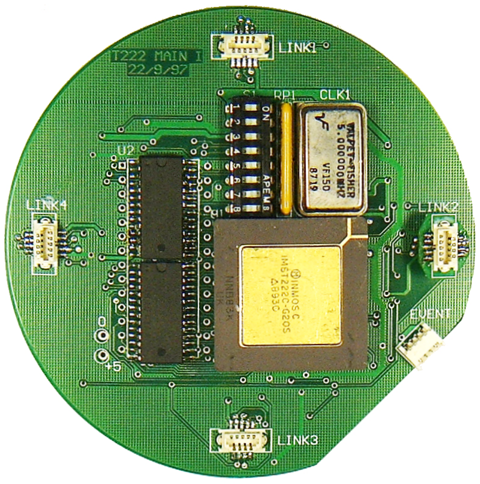
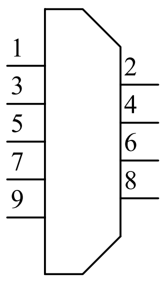


Figure 5 An Agora processor node, based around an Inmos T222 Transputer.

The Agora architecture is a modular hardware design intended for use in small robotics applications [Malmstrom, 2001]. The architecture incorporates the Transputer serial links and the communicating processes idea, and defines a form factor for the physical design of small robots.

Figure 5 shows an Agora node. Each full-size board is an 85mm diameter circle, with four bi-directional Transputer serial link connectors. Power and control signals are also propagated through these connectors, and Agora boards stack together, based on these connectors, to form a column.



1, 2	- VCC
3	- Link out
4	- Error
5	- Link in
6	- Analyse
7	- Reset
8, 9	- GND

Figure 6 An Agora link connector, viewed from above.

Figure 6 shows an Agora architecture link connector, viewed from above the circuit board.

Each peripheral board has one connector with active serial link lines, and three 'pass-through' connectors. Each peripheral board occupies one link channel to connect to a processor node. Each processor node can therefore support up to four peripherals. This is a limitation of the original Transputer implementation, which has only four bi-directional serial links.

Hardware components are encapsulated on separate peripheral boards. The simple serial interface provided by the Transputer links means that each board can behave as though a process is executing in hardware. These "hardware" processes each communicate with a control process executing on a processor node.

Figure 7 shows a possible Agora architecture robot, consisting of a Transputer processor node, a motor controller board, an RS-232 serial board to interface to a desktop PC and a sensor peripheral board. Conceptual hardware processes running on each of the three peripheral boards exchange messages with separate control processes executing on the processor node.

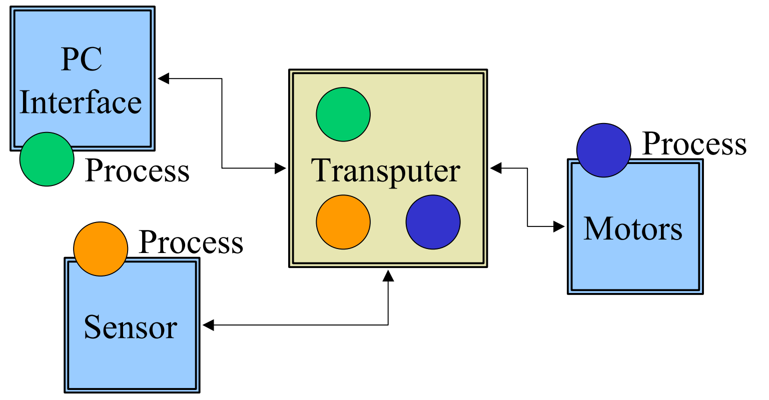


Figure 7 An Agora architecture robot network.

4 The QUTy Mini-robot

The QUTy mini-robot was developed by Kurt Malmstrom for his Mechanical Engineering Ph.D., awarded in 2001. The robot was designed to investigate autonomous robot navigation behaviours with neural networks [Malmstrom, 2001].

QUTy is a small general-purpose research robot, with a two-wheel differential drive, a multi-faceted infra-red sensor and a standard RS-232 serial interface for uploading control software. QUTy was designed using the Agora architecture, and is based around either a T222 or T805 Inmos Transputer node. The robot has only one processor node, but could be expanded with crossover link connectors to incorporate any number of processor nodes and peripherals.

An mentioned above, the Inmos Transputers are now obsolete. Because the Agora system is modular, the processor board can be replaced independently of the existing peripherals. As long as the serial link interface is maintained, the Inmos Transputer can be replaced with a standard von-Neumann processor.

5 Architecture of the Replacement Board

A replacement processor node for the QUTy robot has to meet both the size restrictions introduced by the Agora architecture, and the hardware interface specifications of the Transputer serial links. Aside from these constraints, the new node can upgrade the capabilities of the robot considerably. The Inmos T805 Transputer is a 32-bit processor with a floating-point unit, but runs only at 30 MHz [Inmos, 1992]. Modern processors are significantly more sophisticated than this. Nevertheless, the existing hardware does not need to be modified for use with the new node.

The majority of modern processors have a von-Neumann architecture with a standard microprocessor peripheral bus. Since the replacement node should use off-the-shelf components as far as possible, a standard microprocessor bus should be used for the system bus on the node. The disadvantages associated with a monolithic bus do not apply in this case. The system bus on the node does not need to be modular, and since no off-board connectors are required the width of the bus is not a constraint. The bus is only shared among a very small set of peripherals, so bandwidth limitations are not an issue.

A standard processor is unable to emulate the Transputer serial links without significant computational overhead and specialised software. The link emulation will be carried out by dedicated hardware implemented in a programmable logic device. This device will present a

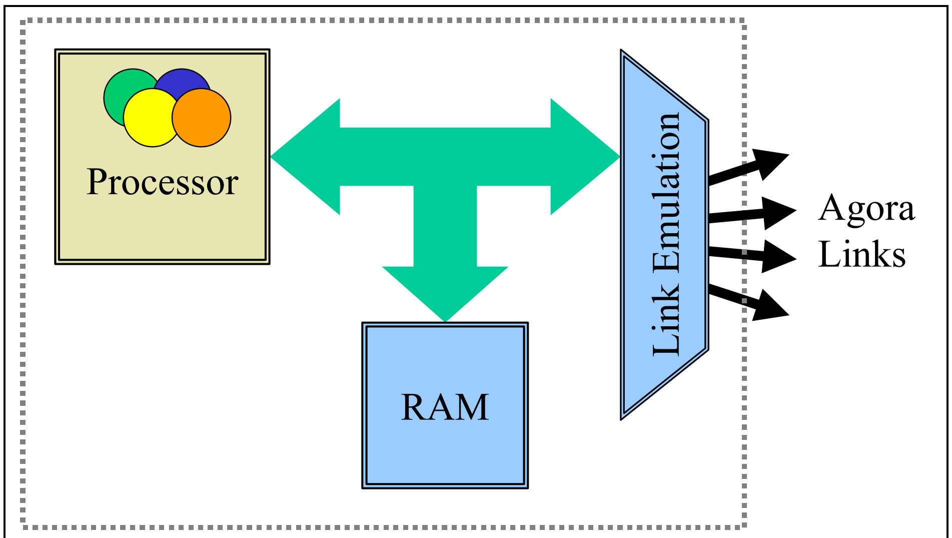


Figure 8 The replacement node architecture.

standard peripheral interface to the system processor, and manage the transmission and reception of data over the Agora serial links. The other peripheral on the system bus will be local memory in the form of RAM. This memory is accessible only via the system bus, and provides code and data storage for the system processor. Figure 8 shows the basic outline of the replacement node.

5.1 Practical Implementation Issues

Due to the size constraints introduced by the Agora architecture and the physical size of the components on the replacement board, an Agora-compliant board would need internal signal layers. These layers present a problem when troubleshooting the board hardware. Internal signal layers are not accessible to logic probes, and so make testing logic levels difficult. We therefore decided to split the development of the replacement node into two stages.

This project would involve the design of a replacement node, and the implementation of the node in the form of a larger, double-sided evaluation PCB (Printed Circuit Board) with no internal signal layers. An adapter PCB would also be produced to interface to the Agora architecture and the existing robot. The design would be independent of the board size, and so could be implemented as an Agora node with no schematic modifications. The adapter board designed for this process is shown in Figure 9.

The second stage, to be completed in another project, would take the completed design for the replacement node and lay out, manufacture and test an Agora-compliant PCB. The second stage would also involve some software development to fully emulate the Transputer functionality.

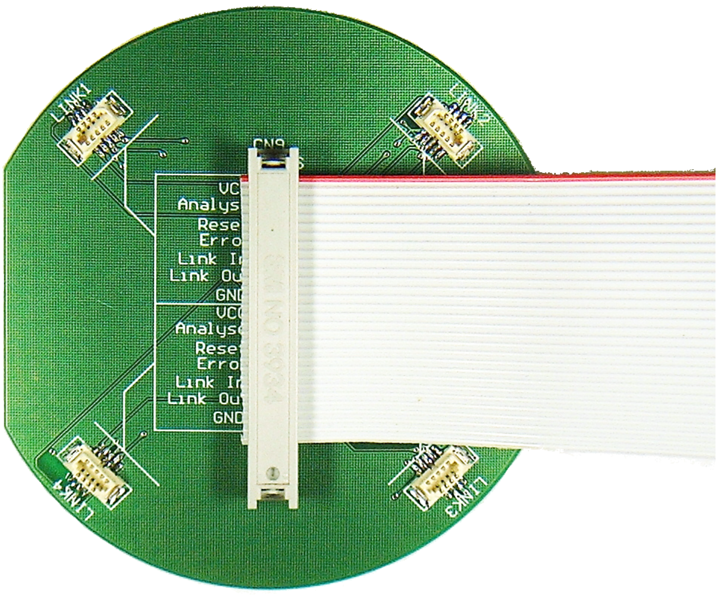


Figure 9 The adapter board developed to interface the Agora architecture links to an IDC ribbon cable.

6 Component Selection

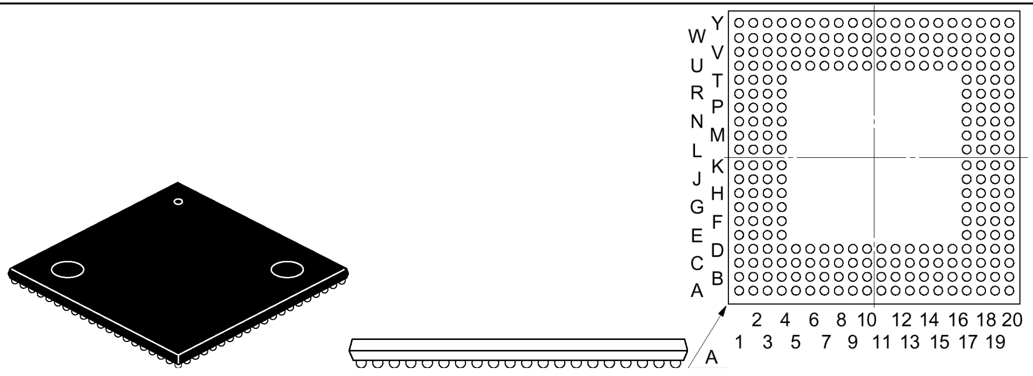
6.1 System Processor

The Hitachi SH7750 (SH4 family) was chosen to replace the Inmos Transputer. The SH7750 is a modern RISC (Reduced Instruction Set Computer) architecture chip, with extensive capabilities [Hitachi, 2000]. The devices operates at an internal clock speed of up to 167 MHz and reaches a peak performance of 300 MIPS (Million Instructions Per Second). The Inmos T805 Transputer has an internal clock speed of 30 MHz and a peak performance of 30 MIPS. The selection of the SH4 was based on a recommendation by Holger Meiners [2001], which identified the SH4 as having an exceptional speed-to-power-consumption ratio.

The SH4 is a low-voltage, low-power component. The processor has many capabilities not required by the replacement node. The SH4 has a built-in PCI bus controller, a PCMCIA card interface and a sophisticated Memory Management Unit (MMU). The chip is designed for embedded computing, and is over specification for control of the QUTy mini-robot. However, it is a relatively cheap component, and satisfies the requirements for a replacement processor.

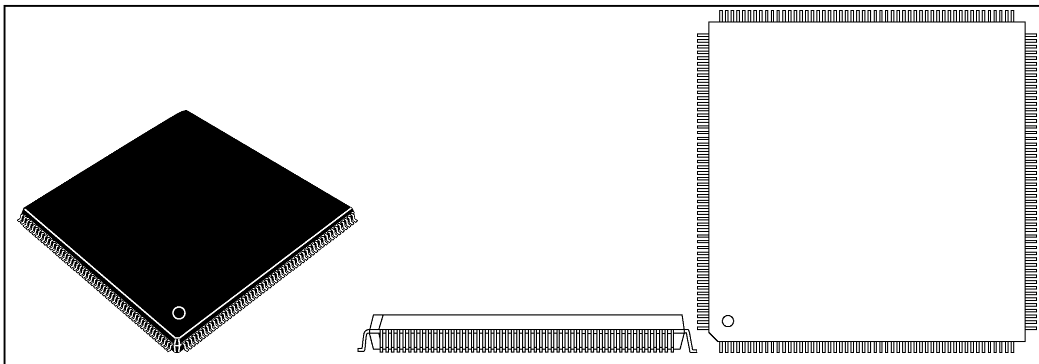
6.1.1 Footprints

The SH4 comes in two packages, a 208-pin Quad Flat Package (QFP) and a 256-pin Ball Grid Array (BGA). Facilities to mount a BGA package were not available, so the QFP-208 package was the only viable option.



Dimensions	27 x 27 mm
Number of pins	256 solder balls
Mounting process	Solder re-flow only

Figure 10 The BGA-556 package.



Dimensions	30 x 30 mm
Number of pins	208 tinned leads
Mounting process	Standard soldering, re-flow

Figure 11 The QFP-208 package.

6.1.2 Support Components

The SH4 processor requires external clock generation, either in the form of a separate oscillation circuit or by using a crystal resonator. The SH4 has internal clock circuitry that manages the crystal resonator, therefore this option is simpler. The processor multiplies the supplied clock frequency to obtain a higher system bus frequency and core clock frequency. A 20 MHz resonator can provide a 120 MHz core clock and a 60 MHz bus clock.

6.2 Serial Link Emulation

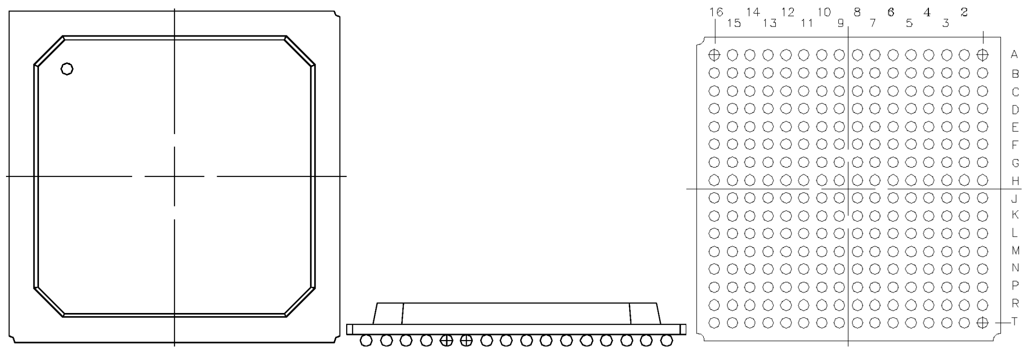
The serial link emulation is based on a programmable-logic device called a Field-Programmable Gate Array (FPGA). FPGAs are described in more detail in section 9, as well as in Clive Hobson's final report [Hobson, 2001].

The Xilinx range of logic devices was chosen because some previous link emulation development by Holger Meiners was based on these devices [Meiners, 2001]. The specific device used by Holger has been phased out by Xilinx. The Spartan-II family of devices was chosen as a replacement. These FPGAs are relatively fast, operating at clock speeds up to 200 MHz. The Spartan-II family has a large capacity in terms of configurable logic space, without the extra features contained in the more advanced members of the Xilinx range. Spartan-II devices are also relatively low-voltage, a desirable attribute for small robotic applications.

The specific device chosen was a Xilinx XC2S200-5PQ208C. This device has 140 user-definable I/O pins and 1176 Configurable Logic Blocks (CLBs).

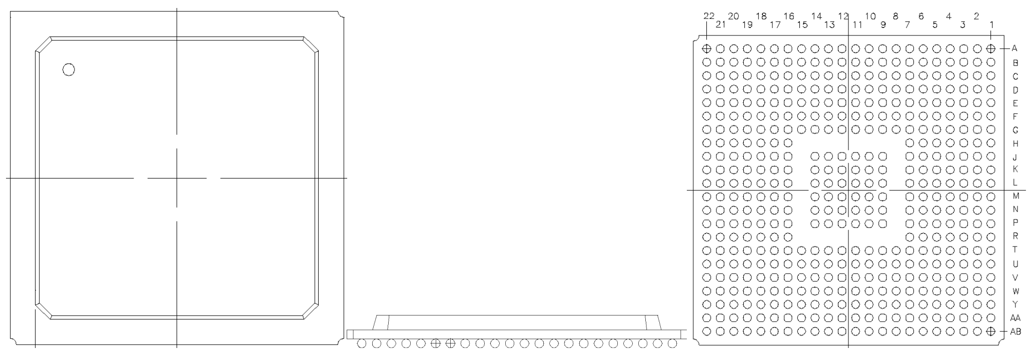
6.2.1 Footprints

The Spartan-II family of devices has a range of packages available. The same selection criteria for the SH4 package apply to the FPGA. No facilities for mounting a BGA package were available, so a QFP package was used. In addition, the number of user-definable I/O pins required by the serial link emulation places a constraint on package selection. The chosen device is only available in one QFP package, the QFP-208 outlined above in Figure 11, which provides sufficient user-definable I/Os.



Dimensions	17 x 17 mm
Number of pins	256 solder balls
Mounting process	Solder re-flow only

Figure 12 The Fine-pitch BGA-256 package.



Dimensions	23 x 23 mm
Number of pins	456 solder balls
Mounting process	Solder re-flow only

Figure 13 The Fine-pitch BGA-456 package.

6.2.2 Support Components

The configuration of an FPGA is volatile, so the component needs to be reconfigured with each power cycle. Xilinx provides a line of in-system Programmable ROMs (PROMs) designed for use with their FPGAs. These devices serially download a configuration bit-stream into an FPGA via a dedicated interface. Both the FPGA and the PROM can be re-programmed via a separate JTAG interface, described in more detail in section 7.4.3.

The PROM chosen was a Xilinx XCV18V02, which has the same voltage requirements as the Spartan-II. This simplifies the design of the board.

6.3 Local Memory and Peripheral Bus

The peripheral bus on the replacement board needs to allow communication between the processor and the serial link emulation logic, as well as between the processor and local memory for execution of code. The bus must also allow independent communication between the serial link emulation and the local memory, to enable direct transfer of link data to and from memory.

6.3.1 Memory Interface Busses

There are several standard memory interface busses. The SH4 directly supports the SRAM, DRAM and MPX interfaces.

6.3.1.1 SRAM Interface

The SRAM (Static RAM) interface bus is by far the simplest. Non-multiplexed address and data busses are combined with chip select, write enable and read enable control lines. Independent data and address busses enable relatively fast transfers between nodes. The small number of control lines makes connections between nodes simple. Static RAM does not require refreshing, so no refresh logic needs to be designed.

6.3.1.2 DRAM Interface

The DRAM (Dynamic RAM) interface bus multiplexes row and column addresses but has a non-multiplexed data bus. DRAM can be set for burst transfer, which transfers a number of consecutive memory locations on the data bus. DRAM needs to be periodically refreshed, which can be managed by the SH4 or by external refresh logic. The DRAM interface has several more control signals than the SRAM interface.

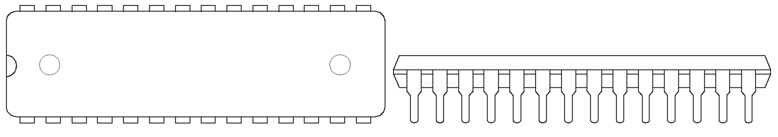
6.3.1.3 MPX Interface

The MPX interface has a multiplexed address and data bus, and is designed for connection to an external memory controller. Multiplexing the address and data bus reduce the bus bandwidth considerably. Extra control signals are also required to enable de-multiplexing.

6.3.2 Choosing A Bus Interface

Since the serial link controller is based in an FPGA, the link controller registers do not require periodic refreshing. There are sufficient I/O pins available on the FPGA such that the address and data busses do not need to be multiplexed. The most obvious access model for the serial link controller is therefore an SRAM interface.

Since the serial link controller must directly access the local memory as well as the processor, the memory must have the same SRAM interface. The final system was designed to have at least 4 MBytes of RAM on the processor node. This amount of memory is not required for evaluation purposes. We chose the largest amount of SRAM that would fit in a simple DIP-28 package (Figure 14).

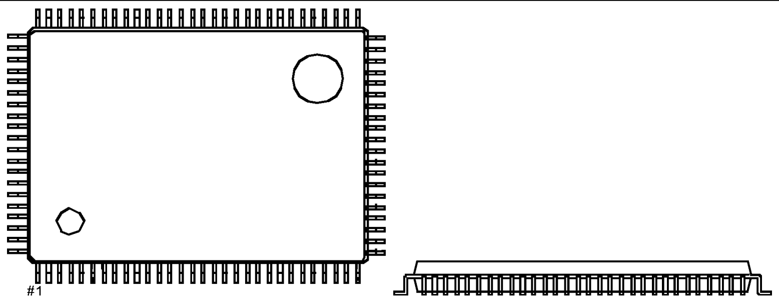


Dimensions	37 x 15 mm
Number of pins	28 tinned leads
Mounting process	Normal soldering

Figure 14 The DIP-28 package.

The Hyundai HY62256BLP-70 has 32 kBytes of SRAM with a data bus with of 8 bits. The access time of the part is 70 ns, which allows a bus speed of 14 MHz. The data rate of the Transputer links requires 2.6 MHz of bus bandwidth per link. With four links receiving simultaneously, this requirement rises to 10.5 MHz. This slower bus speed therefore suffices for evaluation purposes.

For the final system, a Samsung K7B163625M part was chosen. This component contains 2 MBytes of SRAM with a data bus width of 32 bits. The access time is 10 ns, which allows a system bus speed of 100 MHz. The part is available in a QFP-100 surface-mount package (Figure 15).



Dimensions	20 x 14 mm
Number of pins	100 tinned leads
Mounting process	Normal soldering, re-flow

Figure 15 The QFP-100 package.

7 Schematic Design

7.1 Schematic Conventions

Due to the varied voltage supply requirements of the different components of the design, there are several power supply nets on the schematic design. The voltage supplied from the Agora links is a regulated 5V/0V supply, named VCC and GND. The board-wide IO voltage rails are 3.3V/0V, and are named VDDQ and VSSQ. These rails also supply the PROM. The 1.8V/0V rails required by the SH4 are named VDD and VSS. The 2.5V/0V rails required by the FPGA are named VDDF and VSSF. These supply nets are summarised in Figure 16.

Net	Voltage	Used by
VCC	+5V	Board
GND	0V	
VDD	+1.8V	SH4
VSS	0V	
VDDQ	+3.3V	Board IO, PROM
VSSQ	0V	
VDDF	+2.5V	FPGA
VSSF	0V	

Figure 16 The various power supply nets on the schematic design.

The entire set of schematic diagrams is included in appendix A.

7.2 Voltage Regulation

Four separate voltage supplies are present on the board design. The power supply for the entire board is propagated through the Agora architecture links. This is a regulated 5V/0V supply provided by the power supply board on the robot. This supply must be stepped down to accommodate the various components of the design. The voltage regulation is performed by fixed-voltage regulation ICs. Using fixed-voltage regulators rather than

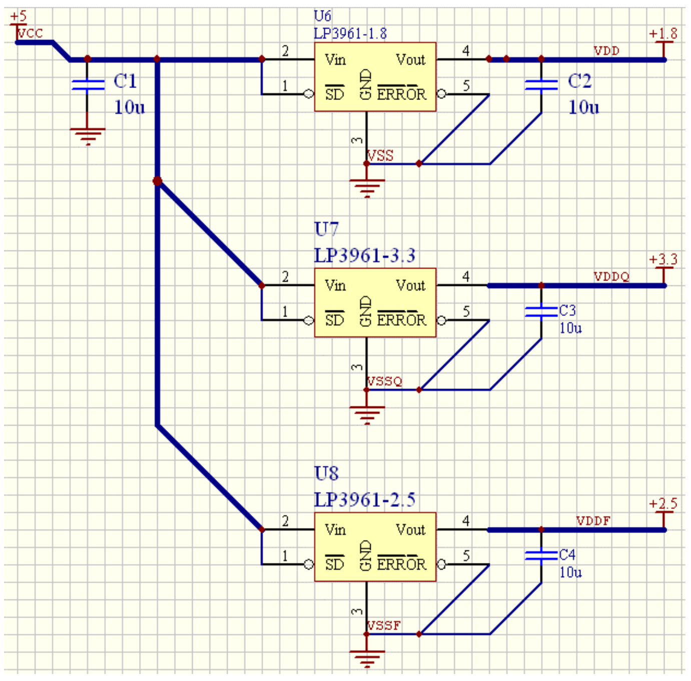


Figure 17 Voltage regulation schematic.

adjustable regulators eliminates the need for trimming circuitry, and ensures accurate supply voltages.

The regulators we chose are National Semiconductor LP3691 low drop-out regulators. This part can supply 800 mA of current. The supply de-coupling circuitry shown in Figure 17 is recommended by the manufacturer [National Semiconductor, 2000].

7.3 System Processor Requirements

7.3.1 Power Supply De-coupling

The high-speed clock circuits inside the SH4 require a very stable voltage supply. To remove any ripple from the supply rails, a de-coupling circuit is used. The circuit in Figure 18 is recommended by Hitachi [2000].

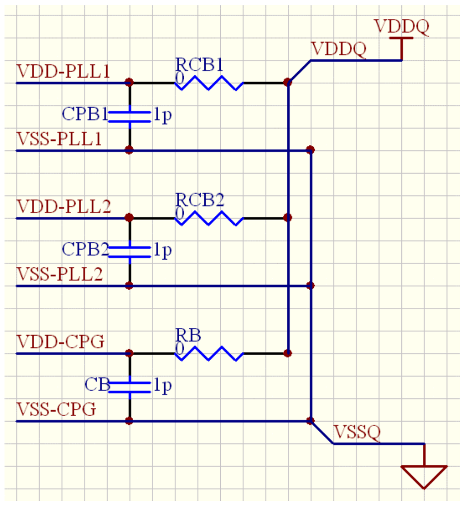


Figure 18 SH4 supply de-coupling schematic.

7.3.2 Reset Configuration

At reset, the SH4 obtains some basic configuration information from the signal levels on a few pins. This information sets the system bus and core clock speeds for the SH4, the system bus width, the endian specification for the chip, the origin of the system clock and the type of interface for the system bus.

These signals are used as mode-setting pins at reset, then revert to other uses when the reset cycle has completed. These other uses are not required in this design, so these pins can be tied either high or low in a fixed configuration.

7.3.3 Resonator Circuit

The 20 MHz crystal resonator is connected directly to GND, and the EXTAL and XTAL pins on the SH4.

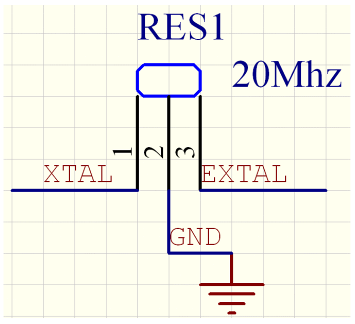


Figure 19 The crystal resonator.

7.4 FPGA and Support Requirements

7.4.1 PROM Interconnections

The PROM uploads the FPGA configuration at power-on, or when the /CF configuration signal is asserted. In the serial configuration mode, the master FPGA supplies a configuration clock to the PROM. The DATA signal on the PROM is the serial data bus, and connects to the DIN signal on the FPGA. No address bus is required, as the PROM is only ever sequentially accessed from the start of the configuration bitstream. No error detection or correction is implemented by the PROM. The configuration clock is sent from the CCLK pin on the FPGA to the CLK pin on the PROM. The /PROGRAM signal on the FPGA connects to the /CF signal on the PROM. The /INIT signal on the FPGA is pulled-up and connected to the /RESET signal on the PROM.

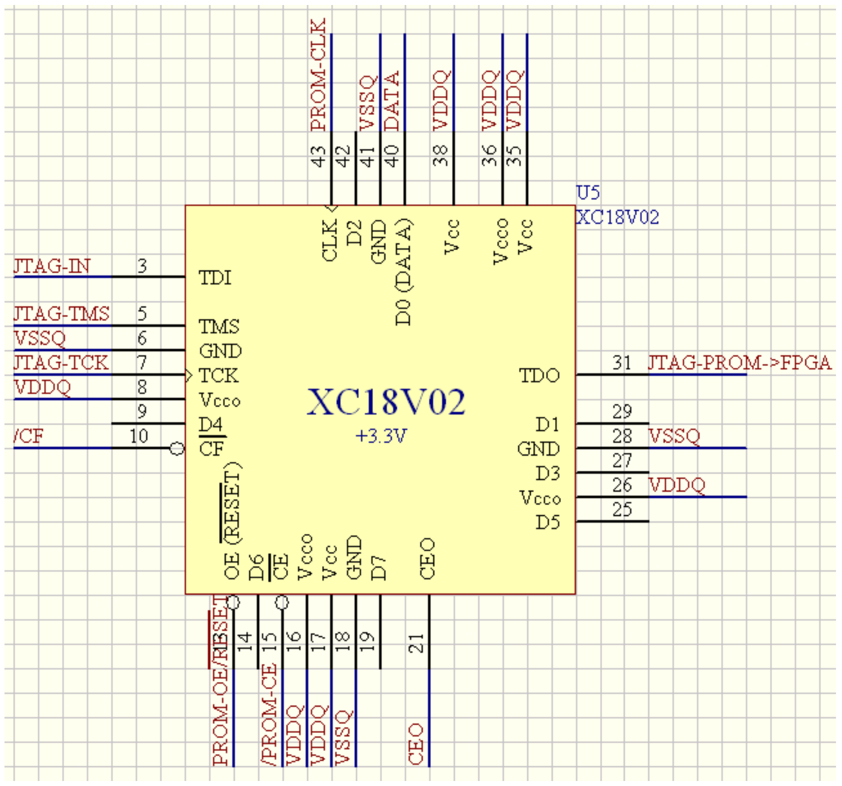


Figure 20 The Xilinx XC18V02 PROM schematic.

7.4.2 FPGA Mode Setting

The mode pins on the FPGA set the configuration mode. In this design, we used a serial configuration mode which uses one data line instead of seven. The total configuration time is less than half a second and configuration occurs only once every power cycle, so the amount of time taken to configure the FPGA is not a design consideration.

To set the serial configuration mode, all FPGA mode pins are tied to VSSQ.

7.4.3 JTAG Port

The JTAG port is used to upload configuration information into the PROM and the FPGA. It is a serial chain bus conforming to the IEEE boundary scan standard (IEEE 1149.1). This standard allows testing of all external signals on the FPGA, as well as uploading and reading back configuration data. TDI and TDO form the serial data chain. TCK provides a clock signal for the synchronous chain. TMS is a control signal that controls the transitions of the JTAG state machines inside the boundary scan devices.

The JTAG port is connected to an external dedicated programming device that interfaces the port to a desktop PC. The JTAG chain is independent of other communications busses on the board.

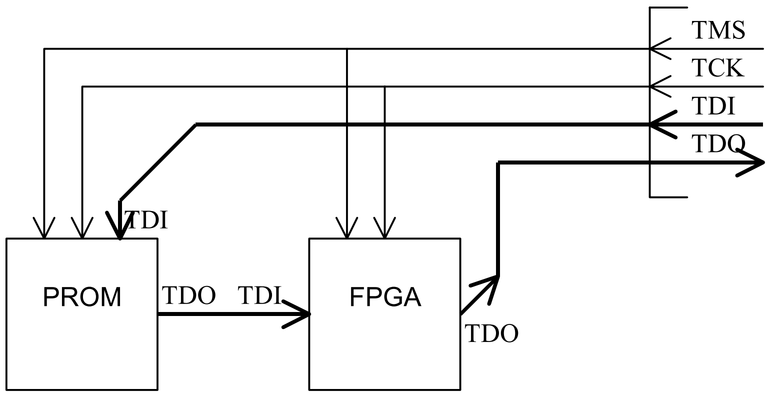


Figure 21 The JTAG chain in the board design.

7.5 Serial Link Controller

The serial link controller has the same interface to the system bus as the SH4. The controller also interfaces to the Agora architecture serial links, and has a few extra control signals. The entire pin-out is given in appendix B.

Signal	Description	Purpose
A25-A0 D31-D0	System address bus System data bus	System bus
GCK3-GCK0	FPGA global clock buffers	System bus clock
/BS /BREQ /BACK	Bus start Bus master request Bus master acknowledge	System bus arbitration
/CS6-/CS0 /WE3-/WE0 /RD	Chip select lines for memory areas 6-0 Write enable lines Read enable	System bus control lines
/CS /OE /WE /READY	Serial link controller chip select Serial link controller output enable Serial link controller write enable Insert WAIT cycle into system bus	Controller slave mode control lines – SRAM interface
IRQ BOOT_HOLD	DMA interrupt request Hold the SH4 in reset while booting from a link	
ANALYSE ERROR LINK1_IN - LINK4_IN LINK1_BUF - LINK4_BUF	No function on this board Agora system error signal Agora serial links – input Agora serial links – to output buffers	Agora architecture control and signal lines

Figure 22 Summary of the serial link controller pin-out.

The serial link outputs are buffered, as the FPGA operates at a 3.3V I/O output logic level, and the Agora architecture uses 5V TTL logic.

7.6 Reset and Boot Logic

Some discrete logic is required to combine the various reset and boot signals present in the design. While the FPGA is being configured by the PROM, the SH4 must be held in reset. The DONE signal from the FPGA indicates when configuration is complete. A signal from the serial link controller, BOOT_HOLD, also must hold the SH4 in reset while a program is downloaded from the serial links to boot the board. There is also an independent reset button on the processor board itself.

The serial link controller has a reset signal to initialise its internal state. This signal must not be asserted while the controller is downloading a program from the serial links. The truth table shown in Figure 23 describes the logic required.

BOOT-HOLD	/DONE	SWITCH-RESET	LINK-RESET	SH4-RESET	FPGA-RESET
0	0	0	0	0	0
0	0	0	1	1	1
0	0	1	0	1	1
0	0	1	1	1	1
0	1	0	0	1	1
0	1	0	1	1	1
0	1	1	0	1	1
0	1	1	1	1	1
1	0	0	0	1	0
1	0	0	1	1	0
1	0	1	0	1	0
1	0	1	1	1	0
1	1	0	0	1	0
1	1	0	1	1	0
1	1	1	0	1	0
1	1	1	1	1	0

Figure 23 Reset and boot logic for the board.

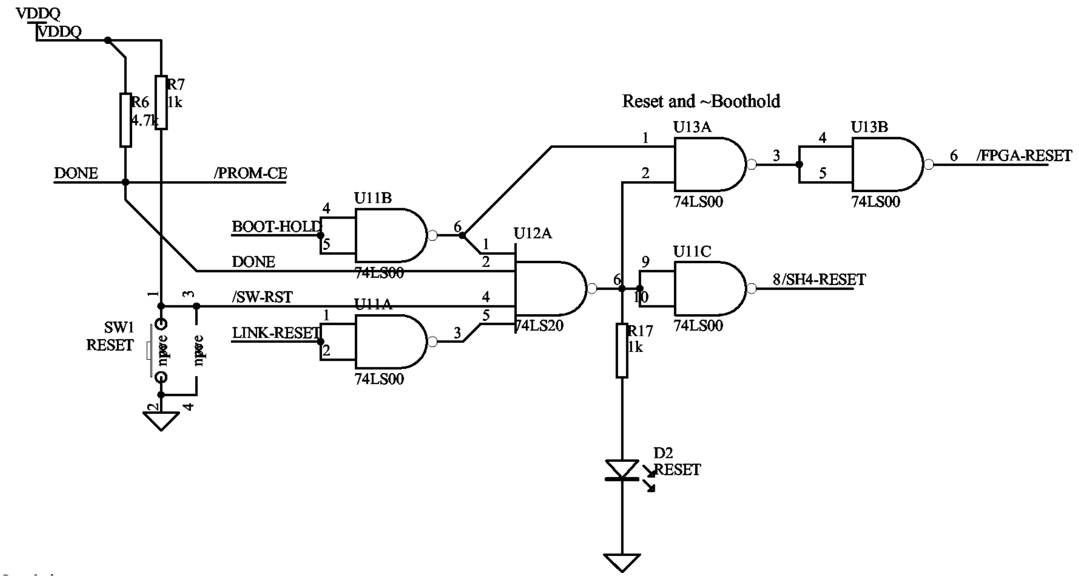


Figure 24 Discrete logic implementation of the reset and boot logic.

The schematic of the discrete logic implementation is shown in Figure 24. The logic was implemented with CMOS NAND gates. These chips are available in small surface-mount packages for the final board implementation.

7.7 SRAM Connections

The RAM connections for the evaluation board are very simple, as it requires only a single chip with an 8-bit data bus. The pin-out for a standard SRAM DIP-28 is shown in Figure 25. Memory area 0 was chosen on the SH4 because the configuration of this memory area can be set at reset with the SH4 configuration pins, described in section 7.3.2. The SH4 provides support for data bus widths from 8 bits to 64 bits. Eight write enable signals are driven by the SH4, /WE0-/WE7, which signal writes to individual bytes of the 64-bit data bus. Chip select signals (/CS0-/CS6) exist for each memory area defined by the SH4. Each memory area encompasses 64 MBytes of memory. The SH4 also supplies a control signal /RD, asserted when the SH4 reads a memory location from the bus.

Connecting a single 8-bit SRAM chip to the SH4 bus is simple. D0-D7 connect to the I/O pins on the SRAM. /WE0 drives the SRAM write enable. /RD connects to the SRAM output enable signal /OE. The SH4 chip select line corresponding to the memory area the RAM is connected to drives the /CS signal on the SRAM.

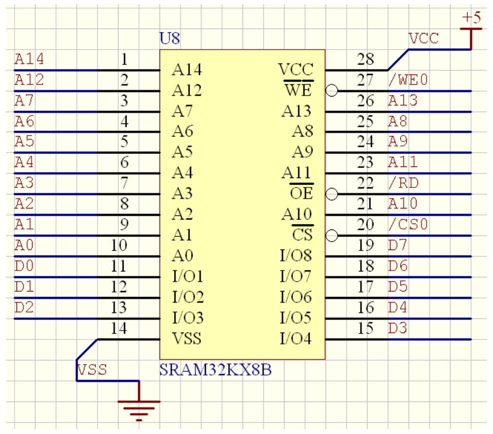


Figure 25 The pin-out for a DIP-28 SRAM chip.

Connecting the 4 MBytes of SRAM on the final board schematic is slightly more complicated. The Samsung 2 MByte component has a data bus width of 32 bits. The device provides four write enable lines, /WEa-/WEd, to access individual bytes of each 32 bit word. The device also provides a global write enable, /GW, which is not used in this design. Three chip select lines allow address decoding for expanded memory depths without requiring extra discrete logic. Two of these components are combined to provide 4 MBytes of memory to the final board.

Each SRAM chip has 18 address lines, which access 512k words of 32 bits each. These lines connect to A2-A20 on the SH4, which thereby addresses blocks of four bytes. A21 on the SH4 is used to decode between the upper and lower 2 MByte banks of memory. When A21 is low, the lower bank is selected. When A21 is high, the upper bank is selected. /OE on the SRAM connects to /RD on the SH4 as before. Since the SH4 provides separate write enable signals for each byte, /WE0-/WE4 connect to /WEa-/WEd on the SRAM.

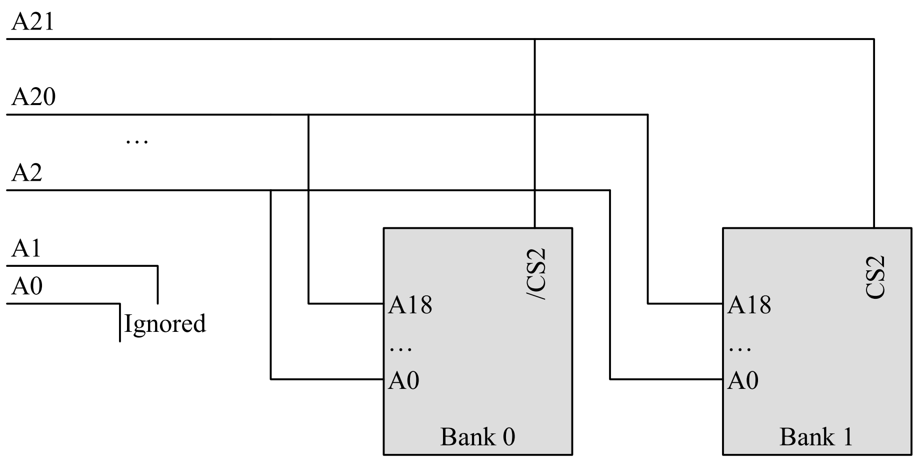


Figure 26 Address bus connections for 4 MBytes of SRAM.

7.8 Debugging Hardware On The Evaluation Board

7.8.1 Test LEDs

A set of five test LEDs was connected to spare IO pins on the FPGA. This assists debugging of VHDL code by allowing selected signals to be displayed.

7.8.2 Auxiliary 5V Supply

An extra voltage regulator was added to the design, to allow the evaluation board to be tested without connecting the robot to supply power. The auxiliary supply is isolated from the board by a jumper, and when enabled can supply power to the Agora boards via the adapter board.

7.8.3 DIP-Switches On SH4 Mode Pins

For the evaluation board, DIP-switches were added to toggle the SH4 mode configuration pins. This enables the testing of various bus speeds and configuration options on the evaluation board. One the final board these mode pins will be tied high or low.

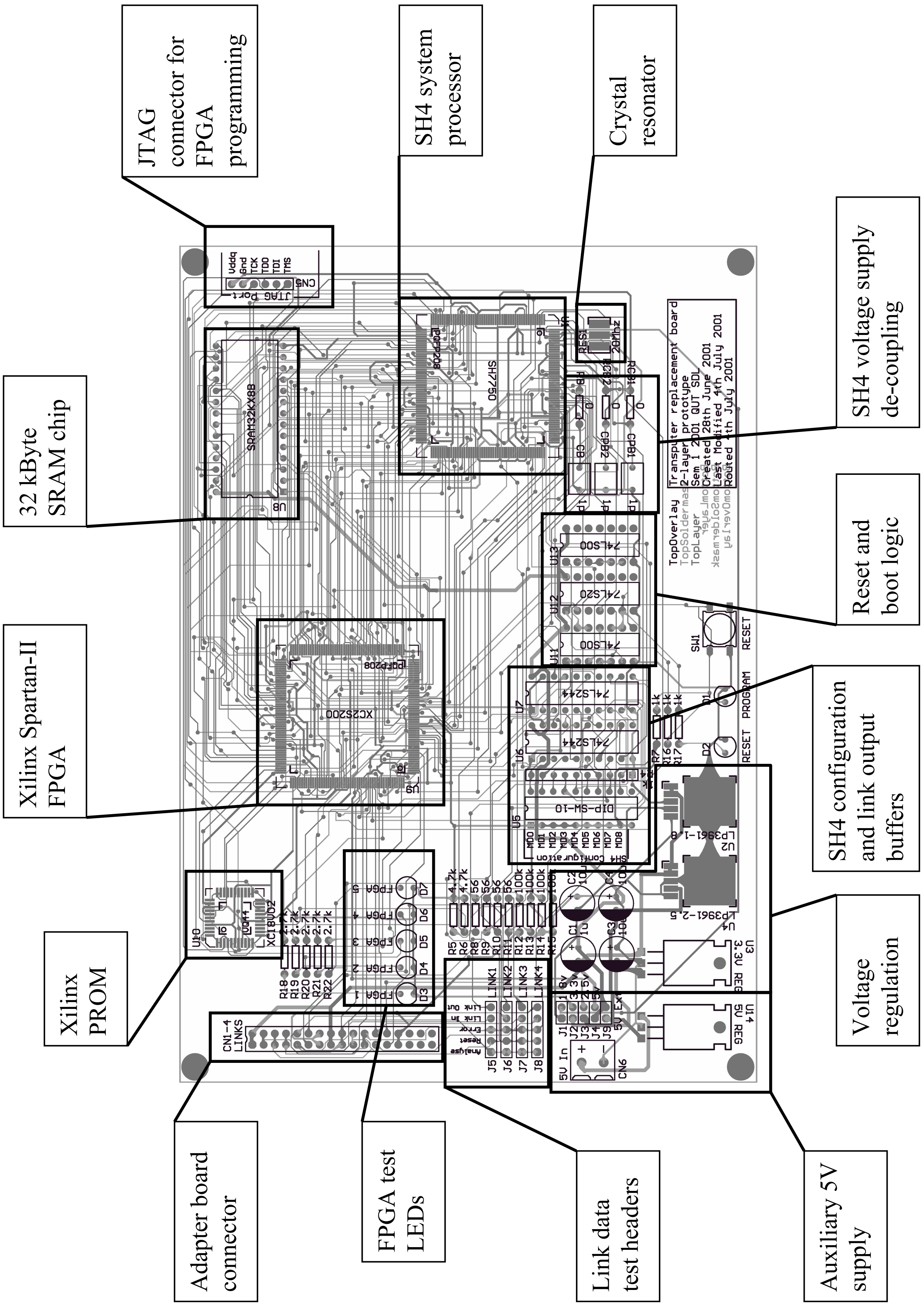
8 PCB Design

The evaluation board fits into an A5 page. There were a number of considerations driving the design of the PCB:

- The evaluation board was designed as a two-layer, double-sided board, to assist with hardware troubleshooting.
- The voltage supply lines were made wider than the signal lines to allow them to carry more current without overheating.
- The voltage supply de-coupling circuits were kept as close as possible to their respective components, to reduce the amount of supply ripple.
- No high-speed lines were routed underneath the resonator or the de-coupling capacitors. This avoids adding high-frequency noise to both the clock and voltage supplies.
- A ground plane was added to both sides of the board, to reduce RF emissions.

On the diagram in Figure 27, the ground planes are not shown for clarity. A larger diagram of the PCB layout is in appendix C.

Figure 27 Annotated layout of the evaluation PCB.



9 FPGAs and VHDL Design

The software design for the serial link controller was performed by Clive Hobson, and is described in more detail in his report [Hobson, 2001]. This section describes FPGAs in general, focusing on the hardware involved and interfacing methods.

FPGAs are volatile programmable logic devices. A basic FPGA consists of an array of configurable logic blocks (CLBs) coupled with a routing matrix and I/O logic for connection to external signals [Xilinx, 2001]. FPGAs operate fundamentally as asynchronous logic. By combining clock signals into the logic, synchronous designs can be implemented. The Spartan-II family of FPGAs also incorporates some fast block RAM and delay-locked loops for digitally eliminating clock skew. A key consideration when selecting an FPGA is the number of user-definable I/O pins available on the device. These I/O pins can be connected to any internal signal net, and so leave the specification of the chip pin-out almost entirely to the designer.

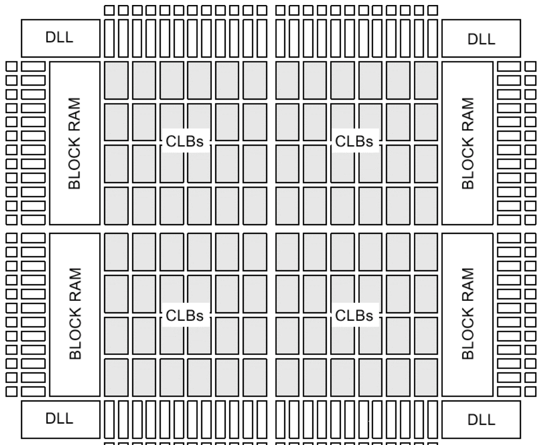


Figure 28 The basic layout of a Spartan-II FPGA.

9.1 Configurable Logic Blocks

Each CLB within the Spartan-II logic matrix consists of four sets of four-input function generators, carry logic and a D flip-flop storage element [Xilinx, 2001]. The function generators are implemented as four-input configurable look-up-tables. These look-up-tables are also capable of functioning as dual-port synchronous RAMs or as high-speed shift registers.

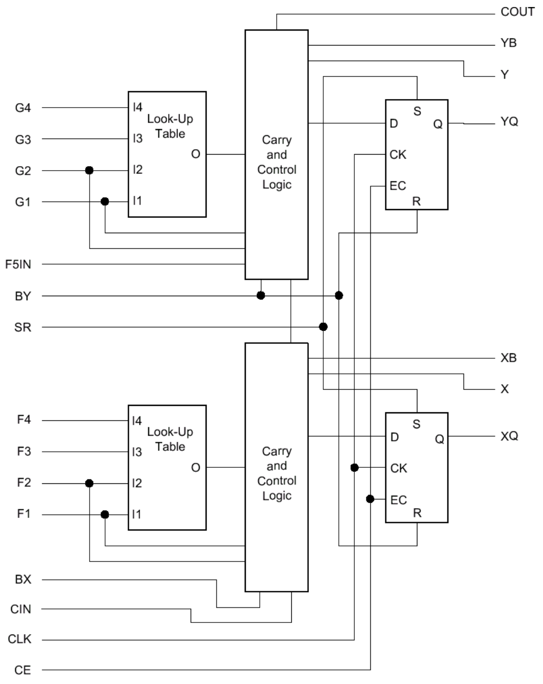


Figure 29 Half of a Spartan-II CLB. Two of these slices make up a single CLB.

9.2 The Routing Matrix

CLBs are connected to each other via various local routing resources. CLBs have internal feedback paths to enable high-speed connections to the same CLB. The global routing matrix (an element of which is shown in Figure 30) provides interconnections between CLBs. The Spartan-II also provides dedicated high-speed routing resources between horizontally-adjacent CLBs, and for connecting CLBs to I/O pads.

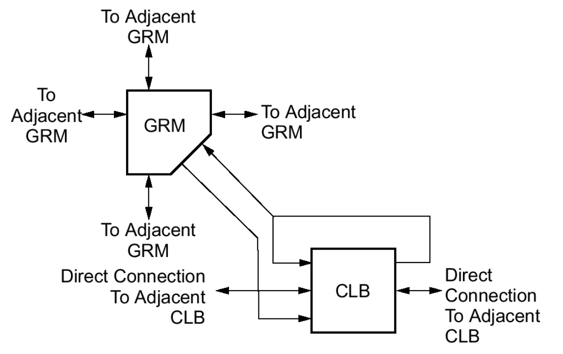

To Adjacent GRM
To Adjacent GRM
GRM
To Adjacent GRM
To Adjacent GRM
Direct Connection To Adjacent CLB
CLB
Direct Connection To Adjacent CLB

Figure 30 An element in the Spartan-II routing matrix.

9.3 The JTAG (Boundary Scan) Interface

The Spartan-II family of FPGAs allows the testing of signals using the IEEE standard 1149.1 boundary scan interface. A dedicated state machine inside the FPGA is controlled by a four signal serial bus. JTAG devices are connected in a chain along this serial bus, and in this way the FPGA and PROM can be configured while soldered to the PCB. This allows for rapid software prototype evaluation and modification.

JTAG is discussed in slightly more detail in section 7.4.3 and in Clive Hobson's report [Hobson, 2001].

9.4 Software Development for FPGAs

The software designed to emulate the serial link controller was written in the **V**ery high speed integrated circuits **H**ardware **D**escription **L**anguage (VHDL). This language allows the specification of circuits at various levels ranging from the schematic level to the connection of the individual gates.

A previous design for the serial link controller existed in VHDL, written by Holger Meiners [2001]. The interface between this design and the system bus needed to be defined and the existing design modified to accommodate the SH4 bus.

This work is described in Clive Hobson's report [Hobson, 2001].

10 Hardware Testing, Errors and Solutions

A few errors in the original schematic design were detected during the hardware troubleshooting phase. None of these errors effect the fundamental operation of the board. The errors, their causes and solutions are outlined here.

10.1 FPGA Testing LEDs

The FPGA test LEDs, added to the design to enable debugging of the VHDL code inside the FPGA, were placed backwards in the schematic. This was due to a option in the Protel schematic software where it is possible to change the appearance of a supply rail connection without changing the actual net being connected to. A “signal ground” symbol was placed, but the connection was to a power net.

This problem was easily solved by flipping the polarity of the LEDs and using inverted logic.

10.2 PROGRAM Status LED

The status LED designed to indicate when the FPGA was being programmed is connected to the wrong signal. The LED is connected to the /INIT signal, which is pulsed by the FPGA at the start of the configuration process. A better choice would be the DONE signal, which is held low until configuration is completed.

This error does not significantly effect the operation of the board, so no solution was attempted apart from correcting the error on the final board schematic.

10.3 JTAG Port Power Pin

The JTAG programming device supplied by Xilinx connects to a standard PC parallel port, but receives its power from the target board. For this reason, an extra pin was added to the JTAG port in the board design, to supply power to the programmer. This power pin is connected to the VDDQ net, at +3.3V. The JTAG programmer requires +5V for operation. This error resulted due to the unclear specifications for the programmer, which do not directly mention the power level required by the device.

The error was corrected for the final board schematic, and remedied on the evaluation board by connecting the programmer to another power pin.

11 Results

Aside from the errors mentioned in section 10 no hardware errors exist on the board, either through schematic errors or through routing errors. The JTAG chain works, and the PROM is able to configure the FPGA. Data has been sent over the Agora links via the adapter board, to control the mini-robot.

Due to the incomplete state of the serial link controller implementation the SH4 peripheral bus has not been fully tested, although the SH4 is known to correctly generate the system clock signal and the bust start (/BS) signal.

The connections from the FPGA to the rest of the board conform to the designed pin-out. The FPGA is able to hold the SH4 in reset. A demo was performed where the FPGA, using a subset of the serial link controller, was able to operate the motor controller board and move the robot in a fixed pattern.

See Clive Hobson's report for a detailed discussion of the serial link controller implementation [Hobson, 2001].

12 Future Work

12.1 Final Board Design and Layout

To enable its use in the Agora architecture, the processor board must be scaled down to meet the Agora form factor. The schematic design for the final board is complete, and is included in appendix D. A preliminary layout for the final board has been performed, and is included in appendix E and shown in Figure 31. This layout uses only surface-mount components, but retains the QFP footprints used on the evaluation board.

The manufacturing requirements for the final board are still to be determined. A routing was successfully performed with six signal layers and four internal power planes, but this many signal layers results in a prohibitively expensive board to produce. The use of BGA package components could be investigated as a way to alleviate routing congestion.

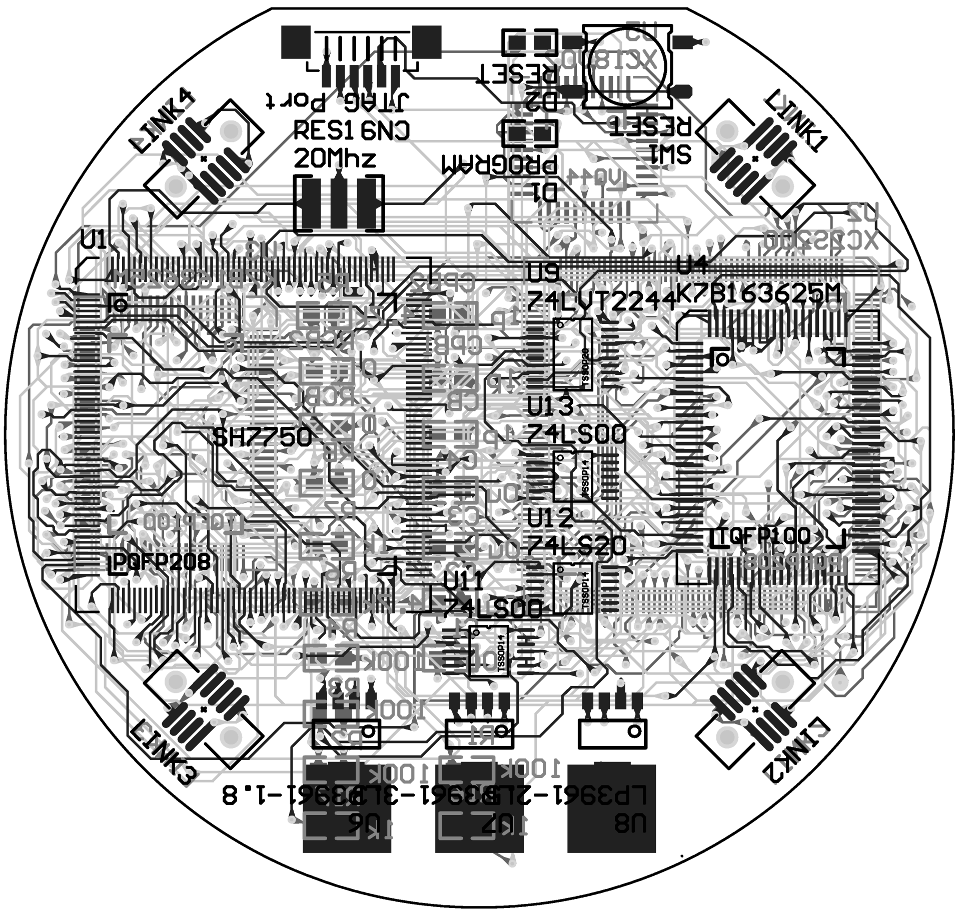


Figure 31 A proposed layout for the final board that conforms to the Agora form factor.

12.2 Software Transputer Compatibility

This hardware design is capable of allowing full software support for Transputer capabilities. However, the design does not include software to provide this support. The SH4 processor does not provide scheduling capabilities or serial link services, both of which are implemented on-die in the Inmos Transputer.

A small operating system needs to be developed to provide these capabilities. Such an operating system should handle all scheduling services, and provide an API to interface processes with the Agora serial links. The operating system should allow interrupt-less systems programming, as discussed in section 2.2.

This operating system could be included on a small ROM for complete transparency, although this would require the shifting of the SRAM to area 1 on the SH4, to place the ROM in area 0 for booting. This would entail minor modifications to the final board schematic design.

12.3 FPGA-Only Implementation

It would be possible to implement the entire system (custom processor and serial link controller) in a single FPGA. This might result in a reduction in core processor speed, but would provide a saving in layout space and system cost. This type of implementation would require choosing an appropriate processor core or designing a new custom core, and adapting the current serial link controller to interface to it.

An interface exists from the serial link controller to the AMBA AHB processor bus. This bus is designed for use internally in processors, and is used by the Motorola M-Core series of soft processors. This type of implementation would also require modification to the schematic design, to interface to the local memory.

See Holger Meiner's thesis for more information on this implementation [Meiners, 2001].

13 References

Hitachi Ltd. 2000, *Hitachi SuperH RISC engine SH7750 Series Hardware Manual* [Online], Available: http://semiconductor.hitachi.com/ [2001, October 24].

Hobson, C. 2001, *A Replacement Processor Node for the Agora Distributed Computing Architecture for Robotics*, Queensland University of Technology, Brisbane.

Inmos Ltd. 1992, *The Transputer Databook*, SGS-Thompson Microelectronics, Italy.

Malmstrom, K. 2001, *Sense-think-act: autonomous behaviour in real mini-robots*, Queensland university of Technology, Brisbane.

Meiners, H. 2001, *Hardware Support for a "Communicating Process Architecture" in Autonomous Control Systems*, University of Paderborn, Paderborn.

National Semiconductor Corp. 2000, *LP3961/LP3964 800mA Fast Ultra Low Dropout Linear Regulators* [Online], Available: http://www.national.com/ [2001, October 25].

Xilinx Ltd. 2001, *Spartan-II 2.5V FPGA Family: Introduction and Ordering Information* [Online], Available: http://www.xilinx.com/ [2001, October 24].

Appendices

A Evaluation Board Schematics

B Serial Link Controller Pin-out

B.1 Serial Link Controller Descriptive Pin-out

Pin Name	Pin	Description	I/O Configuration		Connected to
			Master	Slave	
Address bus					
A25	15	Address bus	O	I	SH4 (System bus)
A24	16		O	I	
A23	17		O	I	
A33	18		O	I	
A21	21		O	I	
A20	22		O	I	
A19	23		O	I	
A18	29		O	I	
A17	30		O	I	
A16	33		O	I	
A15	34		O	I	
A14	35		O	I	
A13	36		O	I	
A12	37		O	I	
A11	41		O	I	
A10	43		O	I	
A9	44		O	I	
A8	46		O	I	
A7	48		O	I	
A6	49		O	I	
A5	58		O	I	
A4	61		O	I	
A3	63		O	I	
A2	67		O	I	
A1	68		O	I	
A0	69		O	I	
Data Bus					
D31	70	Data Bus	O	I	SH4 (System Bus)
D30	71		O	I	
D29	74		O	I	
D28	75		O	I	
D27	81		O	I	
D26	82		O	I	
D25	83		O	I	
D24	86		O	I	
D23	87		O	I	
D22	88		O	I	
D21	89		O	I	
D20	90		O	I	
D19	94		O	I	

D18	96		O	I	
D17	97		O	I	
D16	99		O	I	
D15	101		O	I	
D14	102		O	I	
D13	108		O	I	
D12	110		O	I	
D11	112		O	I	
D10	113		O	I	
D9	115		O	I	
D8	119		O	I	
D7	121		O	I	
D6	122		O	I	
D5	123		O	I	
D4	126		O	I	
D3	127		O	I	
D2	133		O	I	
D1	134		O	I	
D0	135		O	I	
Global clocks					
GCK3	185	Global clock input to FPGA	Always Input		CKIO on SH4
GCK2	182				
GCK1	77				
GCK0	80				
SH4 Bus control lines					
/BS	138	Bus cycle start signal	Always Input		/BS on SH4
IRQ	139	Interrupt request line	Always Output		NMI on SH4
/CS	140	FPGA Chip select slave	HiZ	I	An SH4 chip select line
/OE	141	FPGA OutEnable slave	HiZ	I	/RD on SH4
/WE	142	FPGA WrEnable slave	HiZ	I	/WE0 on SH4
/BREQ	146	Bus Master Request	Always Output		/BREQ on SH4
/BACK	148	Bus Master Ack	Always Input		/BACK on SH4
/CS6	149	Chip Select Area 6	O	HiZ	SH4 Chip select lines. They should map the same memory locations as the SH4 does.
/CS5	151	Chip Select Area 5	O	HiZ	
/CS4	163	Chip Select Area 4	O	HiZ	
/CS3	165	Chip Select Area 3	O	HiZ	
/CS2	166	Chip Select Area 2	O	HiZ	
/CS1	168	Chip Select Area 1	O	HiZ	
/CS0	172	Chip Select Area 0	O	HiZ	
/WE3	173	Write Enable D23-D31	O	HiZ	Memory interface lines on SH4. These should access the same bytes as the SH4
/WE2	174	Write Enable D16-D23	O	HiZ	

/WE1	175	Write Enable D8-D15	O	HiZ	does.
/WE0	176	Write Enable D0-D7	O	HiZ	
/RD	179	Read from peripheral	O	HiZ	
Miscellaneous					
/RESET	180	TLink Interface Reset	Always Input		Reset logic
BOOT_HOLD	181	Hold SH4 in reset	Always Output		Reset logic
/READY	206	WAIT SH4 bus cycle	Always Output		SH4
Transputer Link Lines					
ANALYSE	187	Analyse function	Always Input		All Links
ERROR	188	System error signal	Always Output		Output buffer
LINK1_IN	191	Serial Link Data	Always Input		Link 1
LINK1_BUF	192		Always Output		Output buffer
LINK2_IN	193		Always Input		Link 2
LINK2_BUF	194		Always Output		Output buffer
LINK3_IN	195		Always Input		Link 3
LINK3_BUF	199		Always Output		Output buffer
LINK4_IN	201		Always Input		Link 4
LINK4_BUF	202		Always Output		Output buffer

B.2 Detailed Pin Specification for QFP-208 Package

QFP-208 pin number	Bank	I/O	oth	ref	Defined usage
3	7	1			
Vref (4)	7			1	
5	7	2			
Vref (6)	7			2	
7	7	3			
8	7	4			
Vref (9)	7			3	
10	7	5			
14	7	6			
15	7	7			A25
16	7	8			A24
17	7	9			A23
18	7	10			A33
Vref (20)	7			4	
21	7	11			A21
22	7	12			A20
23	7	13			A19
I/O, IRDY (24)	6		1		
I/O, TRDY (27)	6		2		
29	6	14			A18
30	6	15			A17
Vref (31)	6			5	
33	6	16			A16
34	6	17			A15
35	6	18			A14
36	6	19			A13
37	6	20			A12
41	6	21			A11
Vref(42)	6			6	
43	6	22			A10
44	6	23			A9
Vref (45)	6			7	
46	6	24			A8
Vref (47)	6			8	
48	6	25			A7
49	6	26			A6
Vref (57)	5			9	
58	5	27			A5
Vref (59)	5			10	
61	5	28			A4
Vref (62)	5			11	
63	5	29			A3
67	5	30			A2

68	5	31			A1
69	5	32			A0
70	5	33			D31
71	5	34			D30
Vref (73)	5			12	
74	5	35			D29
75	5	36			D28
77, I, GCK1	5				GCK1, CLKIN
80, I, GCK0	4				GCK0, CLKIN
81	4	37			D27
82	4	38			D26
83	4	39			D25
Vref (84)	4			13	
86	4	40			D24
87	4	41			D23
88	4	42			D22
89	4	43			D21
90	4	44			D20
94	4	45			D19
Vref (95)	4			14	
96	4	46			D18
97	4	47			D17
Vref (98)	4			15	
99	4	48			D16
Vref (100)	4			16	
101	4	49			D15
102	4	50			D14
I/O (~INIT) (107)	3		3		
I/O (D7) (108)	3		4		D13
Vref (109)	3			17	
110	3	51			D12
Vref (111)	3			18	
112	3	52			D11
113	3	53			D10
Vref (114)	3			19	
I/O (D6) (115)	3		5		D9
I/O (D5) (119)	3		6		D8
121	3	54			D7
122	3	55			D6
123	3	56			D5
Vref (125)	3			20	
I/O (D4) (126)	3		7		D4
127	3	57			D3
I/O, TRDY(129)	3		8		
I/O, IRDY (132)	2		9		
133	2	58			D2

134	2	59			D1
I/O (D3) (135)	2		10		D0
Vref (136)	2			21	
138	2	60			/BS
139	2	61			IRQ
140	2	62			/CS
141	2	63			/OE
I/O (D2) (142)	2		11		/WE
I/O (D1) (146)	2		12		/BREQ
Vref (147)	2			22	
148	2	64			/BACK
149	2	65			/CS6
Vref (150)	2			23	
151	2	66			/CS5
Vref (152)	2			24	
I/O (Din, D0) (153)	2		13		
I/O (Dout, Busy) (154)	2		14		
I/O (~CS) (160)	1		15		
I/O (~Write) (161)	1		16		
Vref (162)	1			25	
163	1	67			/CS4
Vref (164)	1			26	
165	1	68			/CS3
166	1	69			/CS2
Vref (167)	1			27	
168	1	70			/CS1
172	1	71			/CS0
173	1	72			/WE3
174	1	73			/WE2
175	1	74			/WE1
176	1	75			/WE0
Vref (178)	1			28	
179	1	76			/RD
180	1	77			/RESET
181	1	78			BOOT_HOLD
182, I, GCK2	1				GCK2, CLKIN
185, I, GCK3	0				GCK3, CLKIN
187	0	79			ANALYSE
188	0	80			ERROR
Vref (189)	0			29	
191	0	81			LINK1_IN
192	0	82			LINK1_OUT
193	0	83			LINK2_IN
194	0	84			LINK2_OUT
195	0	85			LINK3_IN
199	0	86			LINK3_OUT

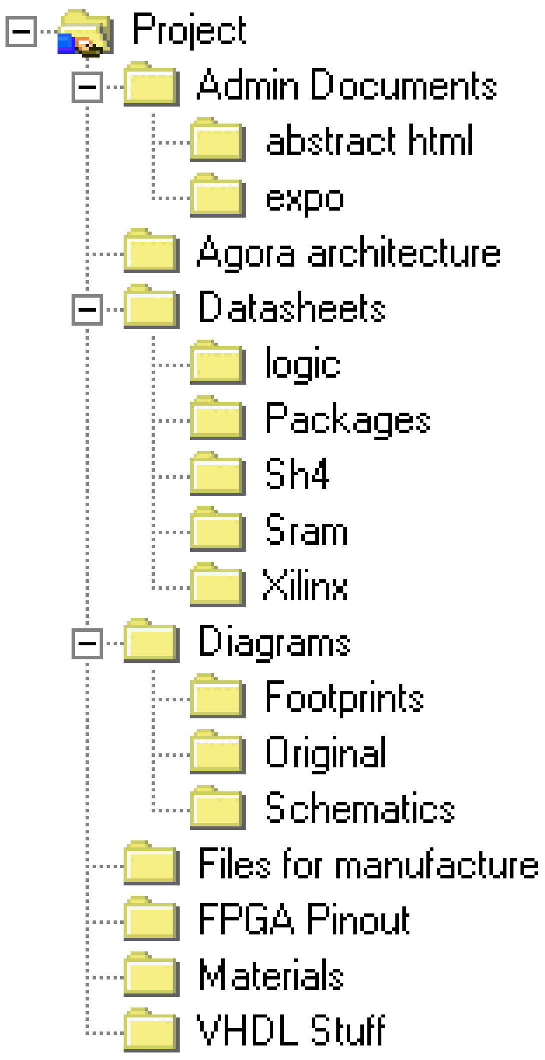
Vref (200)	0			30	
201	0	87			LINK4_IN
202	0	88			LINK4_OUT
Vref (203)	0			31	
204	0	89			
Vref (205)	0			32	
206	0	90			/READY

C Evaluation Board PCB Layout

D Final Board Schematics

E Final Board PCB Layout

F Accompanying CD contents



Main project design database	Project
Project reports, poster, abstract	Admin Documents
Project abstract in HTML format	abstract html
Project expo labels for QUTy modules	expo
QUTy robot schematics	Agora architecture
Collection of component datasheets in PDF format	Datasheets
CMOS logic components and families	logic
Various package specifications	Packages
SH4 documents, including h/ware & s/ware guides	Sh4
Hyundai and Samsung SRAM datasheets	Sram
Spartan-II and PROM datasheets, plus others	Xilinx
Various diagrams in PSP and PNG format	Diagrams
Captures of footprint diagrams	Footprints
Unedited digital photographs	Original
Captures of Protel schematics	Schematics
PCB files exported for manufacture	Files for manufacture
Serial link controller pin-out description	FPGA Pinout
Bills of materials for evaluation board	Materials
VHDL code, help sheets and constraints files	VHDL Stuff

G Selected Component Datasheets

G.1 SH775x (SH-4 Series) SuperH® RISC Processor Overview

G.2 XC18V00 Series of In-System Programmable Configuration PROMs Product Specifications

G.3 Spartan-II 2.5V FPGA Family: Introduction and Ordering Information

G.4 HY62CT08081E Series 32Kx8bit CMOS SRAM

G.5 512Kx36 & 1Mx18 Synchronous SRAM

G.6 LP3961/LP3964 800mA Fast Ultra Low Dropout Linear Regulators

Note: other component datasheets are available on the accompanying CD.