MONASH
AUSTRALIA'S INTERNATIONAL UNIVERSITY

Faculty of Computing and Information Technology

Department of Robotics and Digital Technology

Technical Report 94-5

Caroline Node Interface Specification and Implementors guide.

Cameron Blackwood, Rik Harris, Tony McGregor

May 16, 1995

Enquiries:-

Technical Report Coordinator
Robotics and Digital Technology
Monash University
Clayton VIC 3168
Australia

tr.coord@rdt.monash.edu.au +61 3 9905 3402

Contents

List of Figures

Abstract

Caroline is a low speed, mesh architecture, cell switching network, designed primarily for use in Local Area Networks (LANs), but using some WAN techniques. Some of the design goals are: simplicity, fault tolerance, bandwidth provision by multiple paths and low component cost.

This report describes the Caroline node bus architecture, and the interface information required to design input or output modules that will communicate with a Caroline node using the node bus.

Keywords

Caroline, Network, Hardware, Bus, Specification, Cell Switching, ATM, Asynchronous Transfer Mode, LAN

Chapter 1

Introduction

This document introduces the interface between the *Caroline* bus, and the Input/Output Units. It is intended for use by people designing Input/Output Units, which interface to the *Caroline* node bus. The operation of the node is summarised for reference of the Input/Output Unit designer.

1.1 Other documents.

People unfamiliar with *Caroline* should refer to the *Caroline* introductory paper [1]. Designers wishing to design cards to interface to *Caroline* serial interface should refer to the serial interface paper [2]. More detailed operation of the node bus is described in the node architecture paper [3].

1.2 *Caroline* in brief.

A *Caroline* network consists of a number of interconnected nodes. Each node supports up to 7 interconnections, allowing for multiple, redundant, paths through the network. These paths allow for fault tolerance and high bandwidth capability because of the number of paths between the source and destination.

Transmission of data is accomplished by a connection oriented cell switching method. A connection is established and data is sent in cells across the network over the path established at the time of connection. Because the data is packetised into cells, the network can provide multiple data connections over one physical connection. This allows for virtual circuit cell switching, where the circuit set up at connection time is a path that the cells will take, not a physical connection consisting of exclusively allocated links.

When a cell is received, the port that received it and virtual circuit number in the cell are used to look up the destination port and virtual circuit number. The new virtual circuit number is then placed on the first octet of the cell and the cell is sent to the destination port. This occurs at each node along the path in the network.

1.3 Notes about this document

To reduce confusion regarding active-high and active-low signals, this document will use the terms *assert*, and *deassert* when describing signal behaviour. For an active-high signal, assert means that the signal is to be driven high (+5V), and deassert means that the signal is to be driven low (GND). For an active-low signal (identified by a line above the signal name, for example, $\overline{SIGNAL}$) assert means that the signal is to be driven low (GND), and deassert means that the signal is to be driven high (+5V). If a signal is to be left undriven, it will be specified as either *undriven* or *high impedance.*

1.4 Electronic availability

This paper and other departmental technical reports are available from:

`ftp://ftp.rdt.monash.edu.au/pub/techreports/RDT/`

World Wide Web pages for this project and other technical reports on this project are available from:

`http://www.rdt.monash.edu.au/projects/caroline/`

Chapter 2

Overview of The Caroline Node

This overview is intended to aid developers of Input/Output Units in their understanding of the *Caroline* hardware. For full details of the internal operation of the *Caroline* node refer to [3].

The *Caroline* node is made up of three subsystems: the node subsystem, the node bus and the I/O (Input/Output) subsystem. The node subsystem consists of the Address Resolution Unit, Signal Generation Unit, Central Processing Unit. The I/O subsystem is made up of one or more Input/Output Units. The node bus interconnecting these pieces. figure 2.1 shows how these subsystems are arranged.

2.1 The Node subsystem

Each of the subsystems' functions are described briefly here.

2.1.1 Signal Generation Unit

The Signal Generation Unit generates the control and timing signals required to synchronise all subsystems in the *Caroline* node. The timing signals generated by the Signal Generation Unit are shown in figure 5.2 and figure 5.3.

2.1.2 Address Resolution Unit

The Address Resolution Unit controls the switching of the data cells and translation of the first word of the cell which contains the virtual circuit number. It determines the correct output unit to send the cell to and replaces the virtual circuit of the incoming cell with the correct outgoing virtual circuit number.

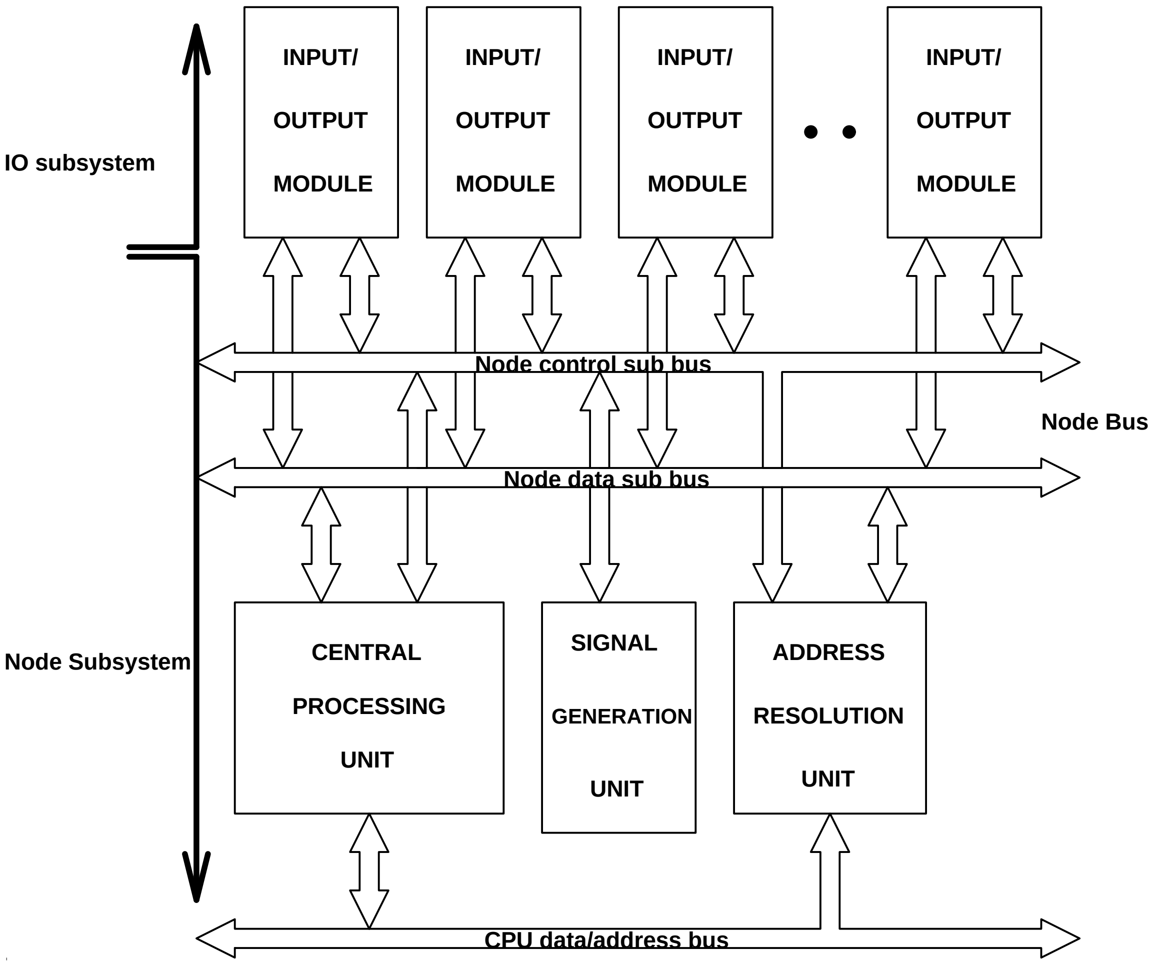

Figure 2.1: *Caroline* node functional modules

2.1.3 Central Processing Unit

The Central Processing Unit handles control cells - cells with the virtual circuit number set to zero. The Central Processing Unit makes routing decisions and participates in the establishment of the virtual circuit. Once the circuit's path through the network has been determined, the Central Processing Units on the nodes involved in that path set the Address Resolution Unit to establish that path. In future releases of the hardware, it is expected that the Central Processing Unit will also handle statistics collection and reporting.

The Central Processing Unit is able to receive control cells from the *Caroline* network. To accomplish this, the Central Processing Unit has an interface to the node bus, similar to the Input/Output Unit interface to the node bus, but the received cell is forwarded to the Central Processing Unit and the data transmitted is generated by the Central Processing Unit. The Central Processing Unit is only a special function Input/Output Unit from the point of view of the node. It receives and transmits cells, just like any other Input/Output Unit.

2.2 Node bus

The node bus connects all of the node's parts together. It is specified in detail in [3] and treated generally in the following sections, especially chapter 3 and appendix B.

2.3 The I/O subsystem

2.3.1 Input/Output Unit

The Input/Output Unit subsystem contains between 1 and 7 Input/Output Units. Each Input/Output Unit has two possible modules, an input module and an output module and must provide one[1] or both of these. Cards providing only one module of this pair cannot take part in the routing currently implemented as routing involves bi-directional transmission[2]. The *Caroline* serial card [2] provides both modules, that is the ability to receive and transmit data.

Examples of possible cards are:

- Printer, an output only device for logging.
- Mains switch, an output only device that allows the remote activation of machinery.
- Camera, an input only device to allow video services.
- Self destruct button, an input only device.
- Audio interface, input and output device, inputting voice and outputting voice.

[1]In this document, each unit in the I/O subsystem will be referred to as a Input/Output Unit even if it has only one of the modules

[2]The nodes that these cards are connected to may provide some of the routing functionality, but this has not yet been standardised by the *Caroline* research group.

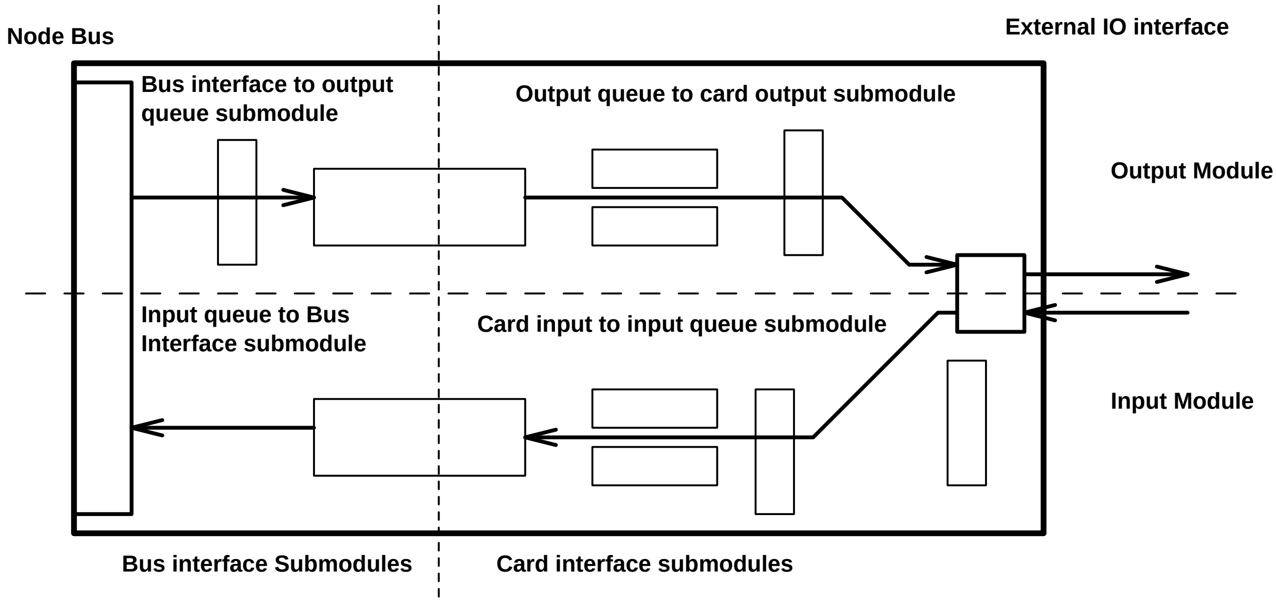

Figure 2.2: Example division of a Input/Output Unit.

- Alien network interface, an input and output device that allows the connection of a *Caroline* network to another network type for internetworking functions.

Each of the input or output modules is made up of two submodules, both of which are required for a module to be useful. The first submodule is the interface to the node bus, that allows the card to send or receive cells through the node bus. The second part is the specific interface provided by that card. This division is shown in figure 2.2.

RULE: Card identification numbers for the input and output modules must be the same.

RULE: Each card identification must be unique.

Each card is given a unique identification number, from 0 to 7, with 0 used by the Central Processing Unit in each node. If input and output bus submodules are both present, they must have the same card number. There must be no more than one module that will respond to a given input module identification number, or output module identification number.

Input/Output Unit bus interface submodules remain inactive until they are selected by their number on either the input selection lines or the output selection lines for input and output submodules respectively; refer to section 4 for more detail. The selection of the bus interface submodules does not affect the operation of the card interface submodules, that is, the operation of the card interface submodules is independent of the node bus interface submodules.

An example of this is the current serial card [2], which has both modules and must therefore have all 4 submodules. The output module is made up of a submodule to receive the cells from the bus and another to translate and transfer them over the serial link. These two submodules comprise the output module. The input module consists of a submodule to receive cells from the serial link and another to transmit the data to the node bus.

Another example is the Central Processing Unit card which has one submodule to receive data from the node bus and another to allow the Central Processing Unit to process that data. The output module takes data from the Central Processing Unit and transmits it to the node via the Central Processing Unit input submodule and bus input interface submodule respectively.

Chapter 3

Signal Descriptions

This chapter describes the states and operations of the signals present on the standard backplane bus. The order of the signals listed does not represent their order on the bus, which is specified in figure B.1. The node bus illustrated in figure 2.1 can be broken down further, as illustrated in figure 3.1.

The standard backplane bus is represented in figure A.1.

Note: this is a simplification, there are more connectors and the bus lines are greater in number and present only on the reverse side of the board on the actual version.

3.1 $\overline{RESET}$

STATUS: required.
RULE: Reset card on assertion of $\overline{RESET}$.

If the $\overline{RESET}$ line is asserted by any module on the bus, all Input and Output modules must be reset, not including the Central Processing Unit. Currently, only the Central Processing Unit will assert the $\overline{RESET}$ line.

The $\overline{RESET}$ line is implemented as a open collector output, to allow many sources to assert it. $\overline{RESET}$ must be asserted for at least T_{reset}, refer to chapter 5.

3.2 Clock (CLK)

STATUS: optional.
OPTION: May be used for timing.

The clock is generated by the Signal Generation Unit. It is 16.384MHz to provide the divide by 8 clock to the Input/Output Unit, as described in [2].

Other special Input/Output Units may ignore or use this clock as they want. It is not vital that

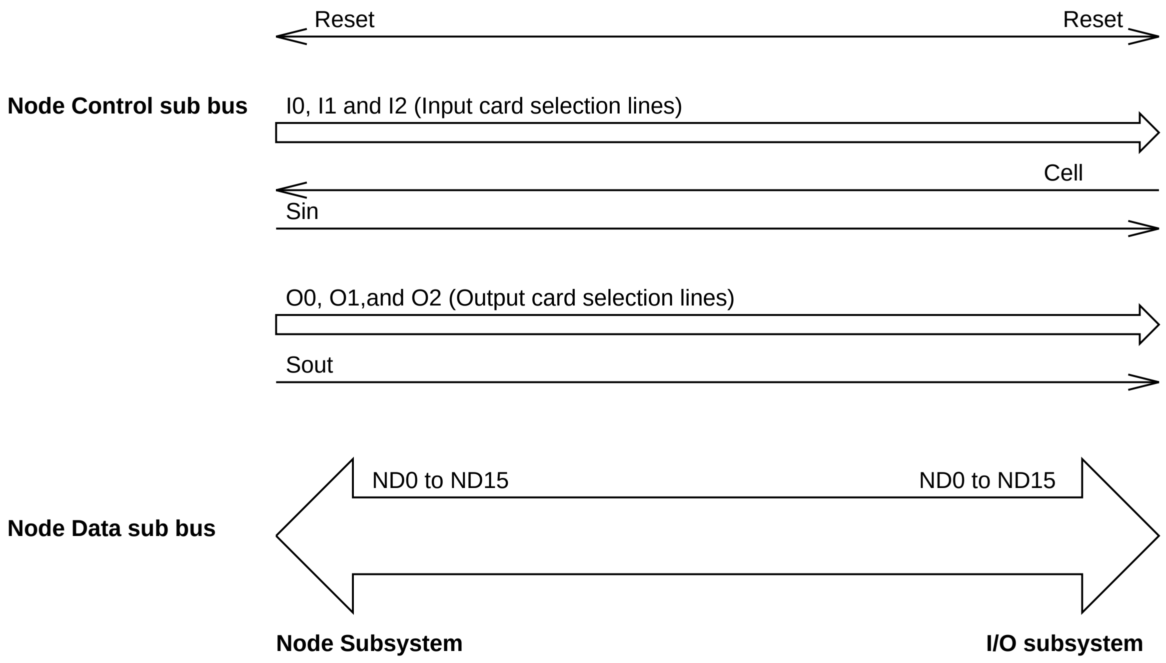

Figure 3.1: Enhanced detail of the node bus.

the Input/Output Unit use this clock to generate its timing signals. The node is designed to allow asynchronous transmission and reception of data, independent of the node bus transfers.

3.3 Input Module Selection $I0$ to $I2$

STATUS: required by input module.
RULE: Activates bus interface.

The input module selection signals are generated by the Signal Generation Unit to poll individual input modules for cells. If an input module has a complete cell in its input queue, and the input module's identification number matches that shown on the input module selection lines, it will assert $\overline{CELL}$ and the Signal Generation Unit will cause a cell transfer to take place. The input module selection lines will then stay unchanged until the cell transfer is complete, at which stage they will move to the next[1] input module. Input/Output Units should not rely on the sequencing of these lines.

3.4 $\overline{CELL}$

STATUS: required by input module.
RULE: Driven when module selected. Asserted if a cell is ready for switching.

The $\overline{CELL}$ signal is used by Input/Output Units to cause the Signal Generation Unit to perform

[1]Note that these lines will not necessarily cycle through 0-7 in the future, to allow the implementation of module priority.

a cell transfer cycle. It must be driven when the card is selected by the $I0$ to $I2$ lines. The $\overline{CELL}$ line is then either asserted or deasserted, depending on the status of the queue.

3.5 Read Selected Input Module S_{in}

STATUS: required by input module.
RULE: When asserted, the selected input module will drive next word of data to the bus.

The S_{in} signal is generated by the Signal Generation Unit, and causes the currently selected input module to output a word from its input queue to the node data bus.

When an input module is selected and S_{in} is asserted, it must drive the data to the node data bus and continue to drive the node data bus until S_{in} is deasserted. For detailed timing, see figure 5.3.

3.6 Output Module Selection $O0$ to $O2$

STATUS: required by output module.
RULE: Activates bus interface.

The output module selection signals are generated by the Address Resolution Unit in response to a virtual circuit lookup. The incoming virtual circuit and the input module identification number are combined to make an index into a lookup table. The result of this table lookup is the outgoing virtual circuit number and the output module identification number. The output module selection lines are held constant until the next cell transfer is required.

3.7 Write to Selected Output Module S_{out}

STATUS: required by output module.
RULE: When asserted, the selected output module will read the next word of data from the bus.

The S_{out} signal is generated by the Signal Generation Unit, and causes the currently selected output module to latch a word from the node data bus, and place it in its output queue. The data on the node data bus is valid while S_{OUT} is asserted- note overlap in figure 5.3.

3.8 Node Data Bus ($ND0$ to $ND15$)

STATUS: required.
RULE: Input modules drive data on these lines, output modules read data from these lines.

The internal node data bus is used to transmit data from an input module to an output module.

Chapter 4

Node Operation

This chapter shall examine the operation of the node using the signals discussed in chapter 3. The input and the output modules operation will be discussed in each of their possible states.

From the Input/Output Unit implementor's view, the node [3]is a cell switch. This switch reads cells from the input modules and writes them to the correct output modules.

4.1 The Input module, transmitting cell to the bus.

SIGNALS USED: input module must use signals $I0$, $I1$, $I2$, $\overline{CELL}$, $ND0$ to $ND15$, S_{IN} and $\overline{RESET}$.

The switching of cells is controlled by the Signal Generation Unit which polls each input module using the input selection lines, $I0$, $I1$ and $I2$, to specify the input module number (0 to 7). An input module is either selected or unselected.

Assertion of $\overline{RESET}$ should reset the input module at any time.

4.1.1 Unselected behaviour

RULE: If Input module unselected then drive no output lines and ignore all signals except $\overline{RESET}$ and $I0$, $I1$ and $I2$.

While the input module is not selected, all output lines are undriven by the card to allow other cards to use the bus. Also, the input module should ignore bus signals, other than the select lines which will select the card, and $\overline{RESET}$ which should always cause the reset of the input module.

Being unselected by $I0$ to $I2$, makes only the input modules bus interface idle. The bus interface should not affect the operation of the input or output modules of the card. Cards are expected to receive and transmit cells in their card interface sections while they are not selected by the node bus - that is, the node interface is idle.

4.1.2 Selected behaviour

Once a card is selected, it transmits the status of its cell queue by driving the $\overline{CELL}$ line to communicate the status of its input queue. The input module should not assert the $\overline{CELL}$ until it has received at least a whole cell, or the implementer can be sure that the rest of the cell will be loaded into the cell queue before the node can read that last octet. It is suggested by the authors that the card should not assert $\overline{CELL}$ until the final word of the received cell has been read to allow a more stable interface and a wider range of timing signals.

RULE: If Input module selected and assert $\overline{CELL}$ then it must output a word each time S_{IN} is asserted. Reset on $\overline{RESET}$.

No cell condition

If the queue is empty, then $\overline{CELL}$ will be driven and deasserted. The Signal Generation Unit will deselect the card and move on to poll the next card.

Cell condition

If a cell is present, then $\overline{CELL}$ will be driven and asserted. The Signal Generation Unit will transfer the words of the cell to the node data bus $ND0$ to $ND15$, by asserting the S_{in} signal once for each 16 bit word.

If a card is selected with $I0$ to $I2$ and the S_{in} signal is asserted, the input module is expected to output the next word in its queue.

The routing step.

This section is included to give the implementor an introduction to *Caroline* routing and a better understanding of the node operation. The Input/Output Unit plays no part in this process.

This first word is loaded into the Address Resolution Unit, which will look up and select the destination output module and virtual circuit, based on the input module number and the virtual circuit number. The Signal Generation Unit will then transfer the first word, with the translated virtual circuit number, to the destination output module. Then, each of the other words of the cell are transferred directly from the input module to the destination output module. Further detail of this process is included in chapter 5.

4.2 Output module, receiving a cell from the bus.

SIGNALS USED: The output module must use signals $O0$, $O1$, $O2$, $ND0$ to $ND15$, S_{IN}, and $\overline{RESET}$.

The switching of cells is controlled by the Signal Generation Unit. The Address Resolution Unit uses the input port number and virtual circuit number to select the outgoing virtual circuit and port. The port is selected using the output card selection lines $O0$, $O1$ and $O2$, to specify the output module number (0 to 7). An output module is either selected or unselected.

Assertion of $\overline{RESET}$ should reset the output module at any time.

4.2.1 Unselected behaviour

RULE: If the output module is unselected then it ignore all signals except $\overline{RESET}$ and $O0$, $O1$ and $O2$.

If the output module selection lines do not match the card's identification number, then the card must ignore data on the node bus, $ND0$ to $ND15$, and the S_{out} signal. The card remains inactive until it is selected.

4.2.2 Selected behaviour

RULE: If the output module is selected then the card will read a word on each assertion of SOUT. The card will reset on assertion of $\overline{RESET}$.

If the output card selection lines match the number of the card, that card must activate the node interface. As stated previously, deactivation of the node interface does not mean the output section of the output module is deactivated. Bus output submodules are selected only when data is transferred to them from the bus. Cards are expected to process those cells in tandem with the operation of the card/bus interface.

When the node interface is active, the output module will read the node data bus lines $ND0$ to $ND15$, into its queue each time S_{out} is asserted, which will be 8 times for a 16 octet *Caroline* cell.

Once the output module has received the data, it is free to process it in its card interface submodules. In the case of the *Caroline* serial interface card [2], the module will start transmitting the cell over the serial link once it has been received.

Chapter 5

Timing

This section describes the timing used to control the transfer of data around the node. The Signal Generation Unit generates most of the signals shown. Chapter 3 describes these signals individually, and Appendix B shows the pin allocations for each of these signals.

The node can be in one of two timing cycles: skipping[1] or switching.

Note: a cycle is 61 ns, as it is derived from a 16.328MHz clock.

5.1 Waiting for a cell

The skipping state of the *Caroline* node is when the polled card has no cell to transfer. After the card is selected, the input module must drive the $\overline{CELL}$ and hold the line, either to assert it if the input module queue has a cell, or deassert it if the queue is empty, no later than T_{sp} after the card was been selected. The $\overline{CELL}$ line must be asserted for at least T_{pr}. If $\overline{CELL}$ is asserted after $T_{sp} + T_{pr}$, or $\overline{CELL}$ is not asserted at all, then the Signal Generation Unit will not transfer the cell. After T_{np} the Signal Generation Unit will deselect the polled module. This timing is shown in figure 5.2. Notice that card 1 has activated $\overline{CELL}$ too late in the cycle.

5.2 Cell Transfer

Once the Input/Output Unit has signalled that there is a cell waiting in its input queue by asserting the $\overline{CELL}$ signal, the Signal Generation Unit initiates a cell transfer sequence. The signals $I0$ to $I2$ are held constant for the duration of the cell transfer. This allows the input module to filter control signals based on these selection signals. The cell transfer sequence is shown in figure 5.3.

Each cell is eight 16 bit words long. The first octet, which contains the virtual circuit number is examined to determine which output module the cell should be sent to, and is modified to include the new virtual circuit number (refer to [1] for more information about virtual circuits).

[1] The skipping state has 2 possible timing cycles, but as the node is not transferring cells, these different cycles do not affect the Input/Output Units, refer to [3].

Symbol	Time	Description	Type
T_{nc}	1 cycle	Minimum time between poll cycle.	
T_{sc}	1 cycle	Maximum time before activation of $\overline{CELL}$ after selection of card.	Must be met
T_{cr}	2 cycle	Minimum time $\overline{CELL}$ must be asserted for if cell present.	Must be provided
T_{so}	1 cycle	Minimum time between selection of output module and writing of first word.	must be met
T_{os}	1 cycle	Minimum setup time for data, that is read time before start of write cycle, data must be stable by the end of this time.	must be met
T_{ol}	1 cycle	Minimum data latch time, data is stable for this time while the write is taking place.	must be met
T_{oh}	1 cycle	Minimum time data is held after write has completed.	must be met
T_{reset}	250 microseconds	Minimum reset time	must reset under this time

Figure 5.1: Timings for the *Caroline* node.

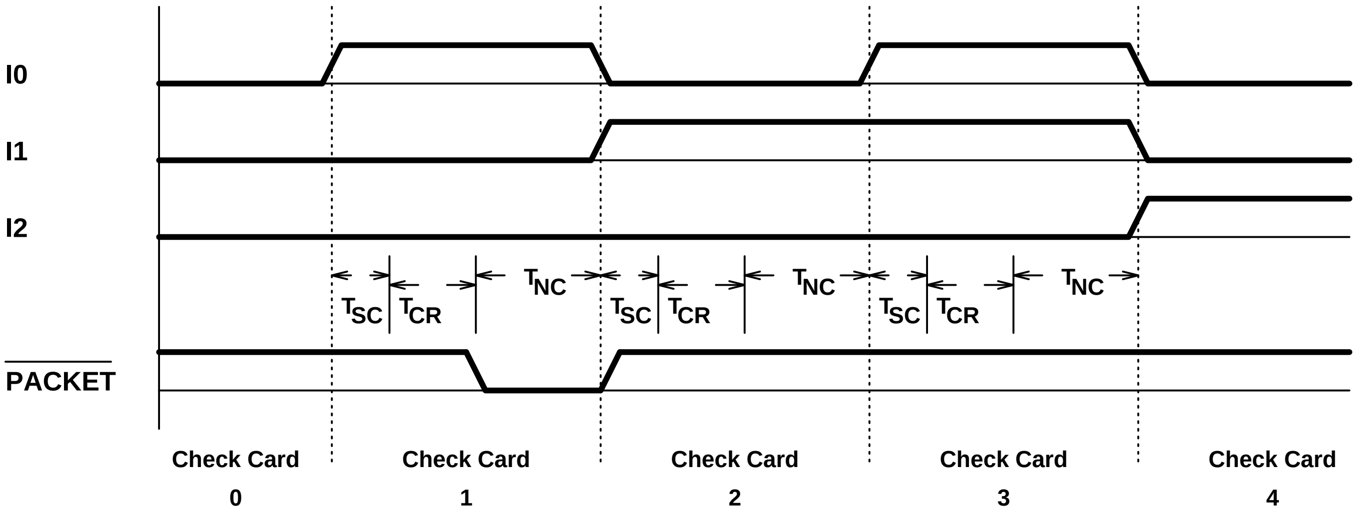

Figure 5.2: Example skipping state timing.

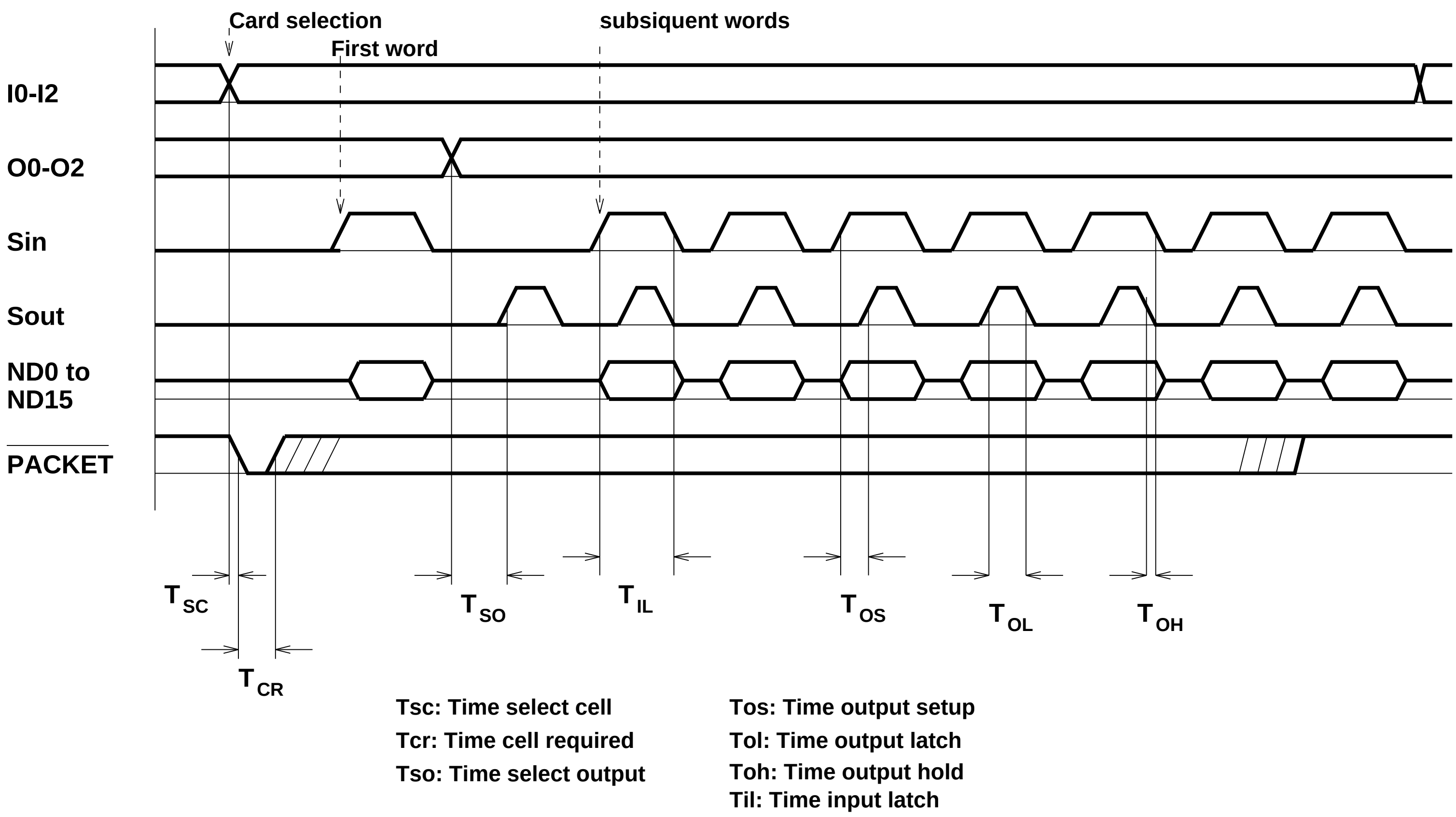

Figure 5.3: Cell Transfer Bus Timing.

Due to the need for the virtual circuit examination, the first word is read from the input module and written to the output module much later (refer to figure 5.3).

The examination and modification of the virtual circuit number is carried out by the Address Resolution Unit during the transfer of the first word of the cell. The Signal Generation Unit first asserts S_{in} to notify the input module to present a word on the node data bus ($ND0$ to $ND15$). The Address Resolution Unit latches this word and performs a table lookup based on the incoming virtual circuit number and the input module identification number ($I0$ to $I2$). When the S_{in} signal is deasserted, the input module must cease driving the node data bus. After the Address Resolution Unit has performed the lookup, the reconstructed first word with the outgoing virtual circuit number is output on the node data bus. At the same time, the Address Resolution Unit selects the appropriate output module by changing the output module selection signals ($O0$ to $O2$). The S_{out} signal then notifies the selected output module that the data on the node data bus is valid and should be stored in the output queue.

The Signal Generation Unit controls the transfer of the remaining 7 words by asserting S_{in} to notify the input module to present a word on the node data bus, followed by S_{out} to notify the output module that the data on the node data bus is valid and should be placed in the output queue then deassert S_{out} then deassert S_{in}. The output module selection lines ($O0$ to $O2$) are kept set to the selected output module until the next cell is transferred. At the end of this sequence, the Signal Generation Unit selects the next input module in the sequence, and returns to waiting for a cell - the skipping state.

Note that if an input module has two or more cells in its input queues, only one is sent. Other cells are sent when the Signal Generation Unit gets back to that module in the polling sequence in subsequent polls.

Chapter 6

Schematics for Bus Interface Sub-Modules

This chapter contains outlines and schematics for an example bus interface submodules used in the initial design of the *Caroline* hardware. These submodules provide the interface to the *Caroline* node bus.

6.1 Input Module Bus Interface

Figure 6.1 shows a typical input module interface to the node bus. The operation of this submodule is to take data from the queue in its FIFOs and transfer it over the bus. When the card interface submodule has received a cell and stored it in the FIFO, it sets the flipflop. This will cause the assertion of the $\overline{CELL}$ line next time the card is polled by the node. This submodule will then read the data from the FIFOs to the bus and clear the flipflop. The submodule then waits until the card interface submodule has loaded the next cell.

The GAL equations for this schematic are listed below in figure 6.2. $\overline{ISEL}$ is an active low signal and is low only when the $I0$ to $I2$ lines match the $SW0$ to $SW2$ lines, that is the polled card is the same as the card number. $\overline{ToBus}$ is also active low and is only active when S_{in} is low and $\overline{ISEL}$ is active. $\overline{ISEL}$ also activates the output enable on the $\overline{CELL}$ line, which is derived from the status signal from the FIFO.

6.2 Output Module Bus Interface

Figure 6.3 shows a typical output module interface to the node bus. When the $O0$ to $O2$ selection lines match the card number, then S_{out} causes data to be written into the FIFOs.

The GAL equations for this schematic are listed below in figure 6.4. Note that $\overline{FromBus}$ is an active low signal, and is low only when the $O0$ to $O2$ lines match the $SW0$ to $SW2$ lines and the S_{out} is active.

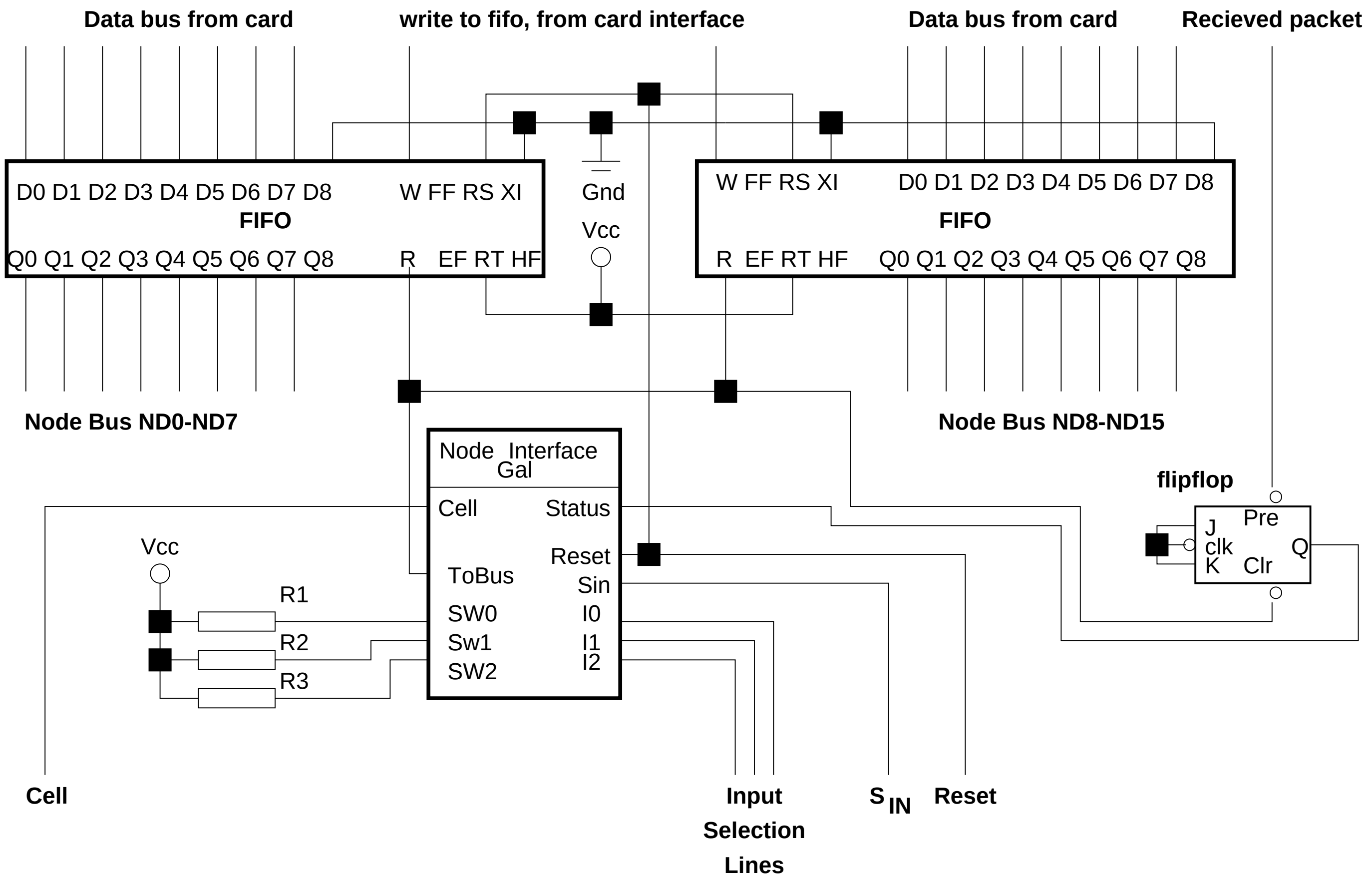

Figure 6.1: Input Module Schematic

```
isel := /sw0 * i0
    + sw0 * /i0
    + /sw1 * i1
    + sw1 * /i1
    + /sw2 * i2
    + sw2 * /i2

to_bus = /sin + isel
to_bus.oe=vcc

cell = /status
cell.oe = /isel
```

Figure 6.2: Input Module GAL

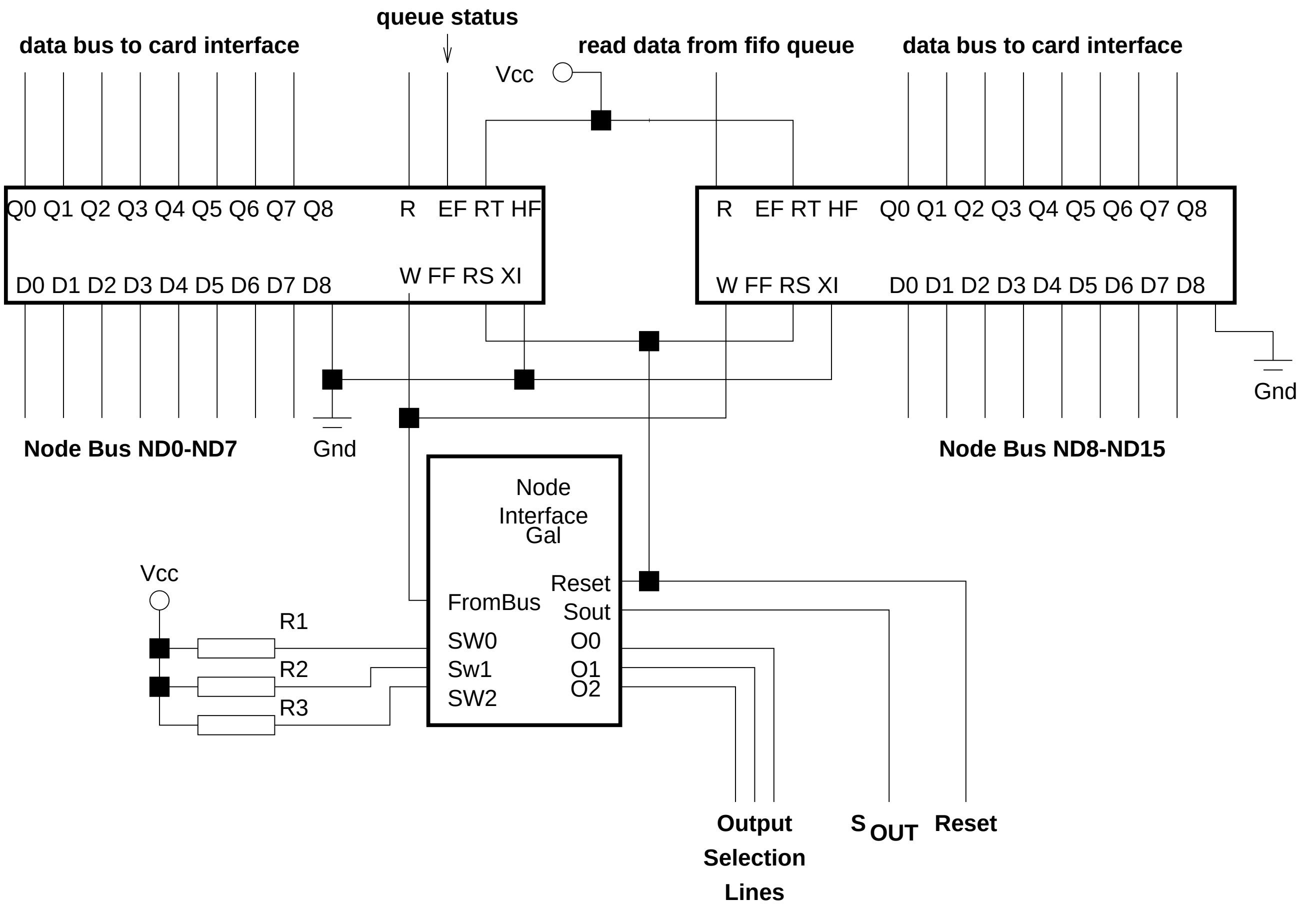

Figure 6.3: Output Module Schematic

```
from_bus := /sout + /sw0 * o0
    + sw0 * /o0
    + /sw1 * o1
    + sw1 * /o1
    + /sw2 * o2
    + sw2 * /o2
```

Figure 6.4: Output Module GAL

Chapter 7

Mechanical Specification

This section is included to ensure all components of a *Caroline* system can be physically assembled. The mechanical specifications are very similar to those of the VMEbus [4].

7.1 Board Dimensions

The standard board for an Input/Output Unit is 100mm × 160mm × 1.6mm thick. A 603-2-IEC-C096-M connecter on the smaller edge provides connection to the backplane. See figure 7.1. The prototype allows a board of dimensions 233.35mm × 160mm × 1.6mm thick with a single 603-2-IEC-C096-M connector as shown in figure 7.2. This card size will not be supported in the final specification.

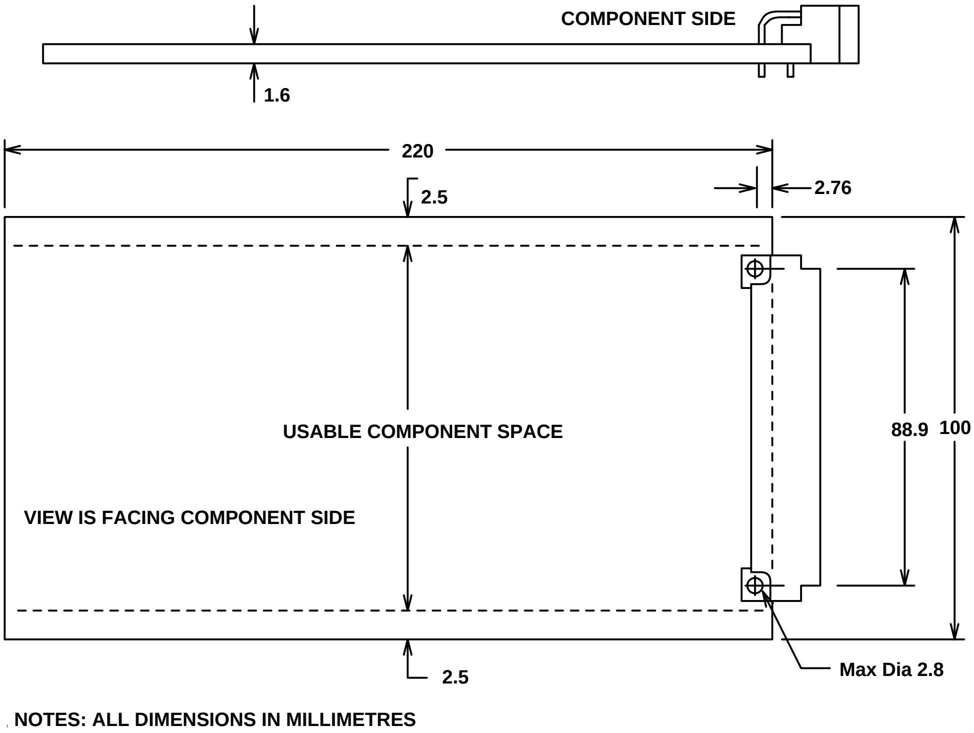

Figure 7.1: Standard Board Dimensions

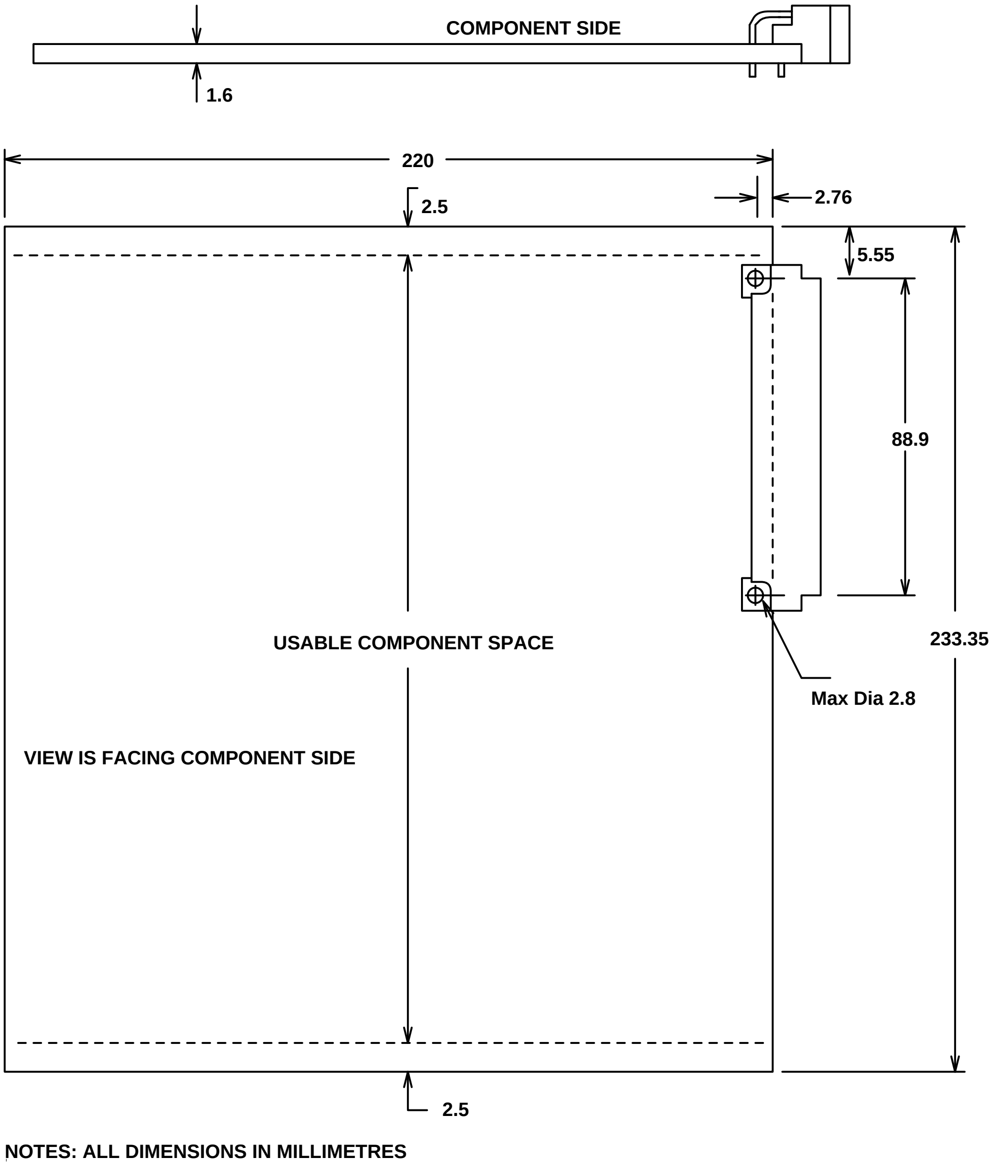

Figure 7.2: Extended Prototype Board Dimensions

Appendix A

Bus Electrical Specification

A.1 Ground and +5

A supply voltage of between 4.8V and 5.2V is required to drive the *Caroline* node. This is supplied to each card.

A.2 Signal load

Each signal from the node bus should drive at maximum only one input on each card. Signals that are required in multiple places on a card should be buffered as needed.

Each output to the bus should be able to drive 8 CMOS logic inputs and the termination discussed below.

A.3 Termination

The node bus termination groups provide termination and filtering to some of the node bus signals, in the form of 8 termination groups, four at each bus end, as illustrated in figure A.1.

Each termination group consists of 2, 9 pin sockets for resistor packs and 8 capacitor connectors, as illustrated in figure A.2. Together, this provides for optional pullup to Vcc and/or high frequency filtering.

Bus signals $a1$ to $a23$ are pulled up and all signals, except $a5$, have high frequency filtering. The pullup is 4.7k ohm and the capacitor value is 120pF. $A5$ has no high frequency filtering because it is a high frequency signal which requires as little signal modification as possible.

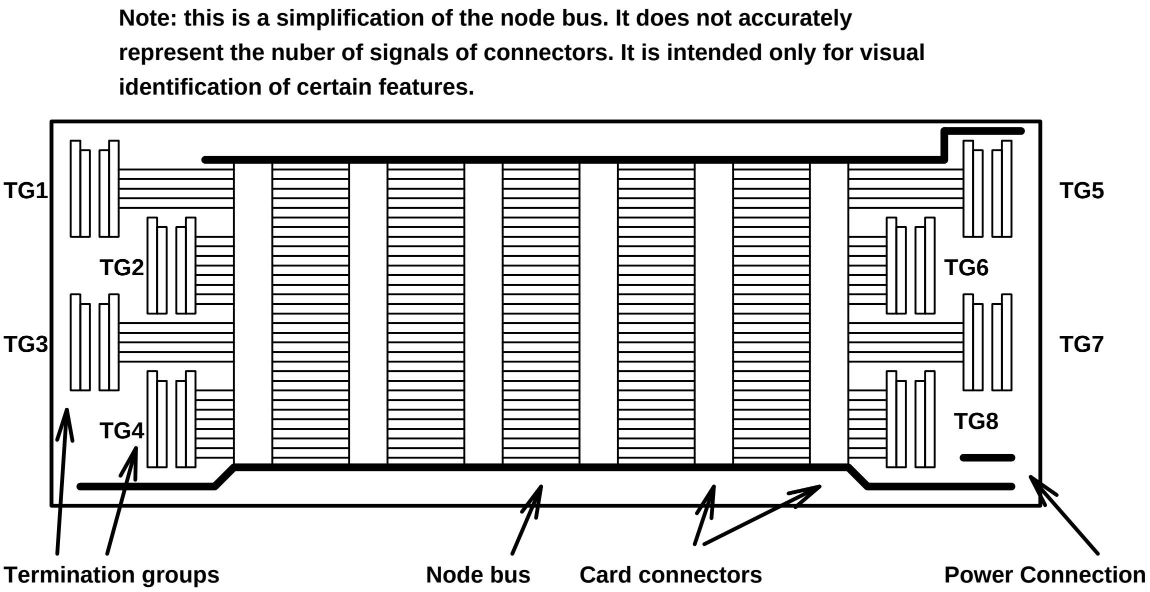

Figure A.1: Node bus and termination.

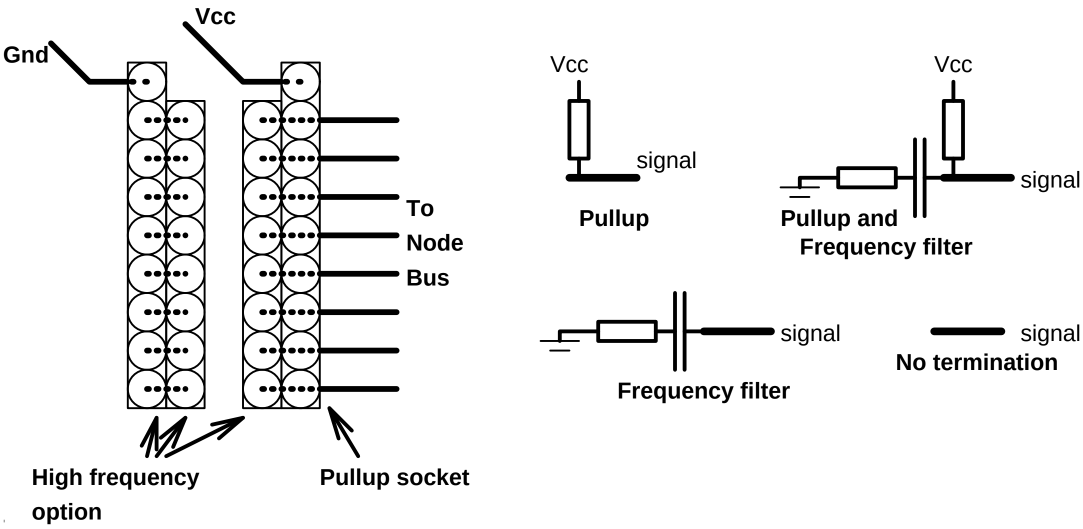

Figure A.2: Termination group connections.

Appendix B

Backplane Pin Allocations

Signal	Designation	Function
a1	GND	
a2	NC	
a3	$\overline{CELL}$	cell status
a4	$\overline{RESET}$	reset node
a5	Clk	clock for node
a6	NC	
a7	$ND0$	
a8	$ND1$	
a9	$ND2$	
a10	$ND3$	
a11	$ND4$	Internal
a12	$ND5$	Node
a13	$ND6$	Data
a14	$ND7$	Bus
a15	$ND8$	
a16	$ND9$	
a17	$ND10$	
a18	$ND11$	
a19	$ND12$	
a20	$ND13$	
a21	$ND14$	
a22	$ND15$	
a23	$I0$	Input
a24	$I1$	Module
a25	$I2$	selection
a26	$O0$	Output
a27	$O1$	Module
a28	$O2$	Selection
a29	S_{in}	read data from input submodule
a30	S_{out}	write data to output submodule
a31	NC	
a32	+5	

Figure B.1: Standard Backplane Pin Allocations

Bibliography

[1] Cameron Blackwood, Rik Harris, and Anthony McGregor. Introduction to the Caroline Network. Technical Report 95-unpublished, Monash University, Department of Robotics and Digital Technology, 1995.

[2] Cameron Blackwood and Rik Harris. Caroline Serial Interface Specification and Implementors guide. Technical Report 95-unpublished, Monash University, Department of Robotics and Digital Technology, 1995.

[3] Cameron Blackwood and Rik Harris. The Caroline Node Architecture. Technical Report 95-unpublished, Monash University, Department of Robotics and Digital Technology, 1995.

[4] Signetics. *VMEbus Specification Manual Rev. B.* Phillips., 1982.

Appendix C

About the Authors

Cameron Blackwood, Rik Harris and Tony McGregor have been involved in development of the *Caroline* project at Monash since it began in 1990. At the time of writing, Cameron is working on a PhD in *Caroline* design and management, Rik has almost completed his masters by research in routing methods and Tony is a senior lecturer in the department and supervisor of Cameron and Rik.

Email: cameron.blackwood@rdt.monash.edu.au
Email: rik.harris@rdt.monash.edu.au
Email: tony.mcgregor@rdt.monash.edu.au

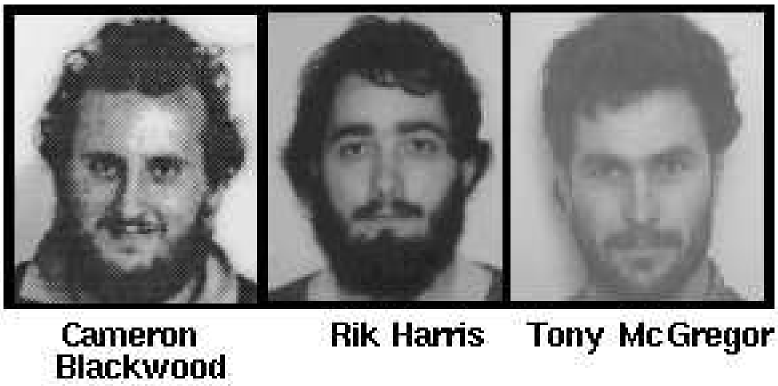

Figure C.1: The Authors.