Engineering Notes: 16-Bit I / O Bus Controller

Matthew Rothlisberger

2019-06-24

1 Overall Design Goals

This system is one part of a project to develop a custom 16-bit computer constructed solely using 74-series discrete logic components, comprising a processor and necessary peripherals. We aim to construct a computer comparable to past systems such as the IBM PC or the Apple II, which can run a functional operating system, a terminal, and even software such as games. The bus controller is one of the six systems included within the processor.

1.1 Processor Subsystems

Control The control unit is the brain of the processor, sequencing the flow of information through all other components and ensuring that instructions are executed properly and that operations take place in order.

ALU The Arithmetic Logic Unit, the heart of the processor. Performs all mathematical operations, with hardware capability for addition, subtraction, negation, and multiplication.

Registers The register file consists of eight 16-bit registers to hold data in use by the processor: 5 general purpose shift registers, 1 register permanently set to zero, 1 stack pointer, and 1 presettable up / down program counter.

Comparator The comparator system allows the processor to execute conditional branching instructions, by indicating whether one value is greater than, less than, or equal to another.

Buffer The 16-bit buffer allows data to be transferred between the processor's two internal buses as needed, facilitating data transfer between registers.

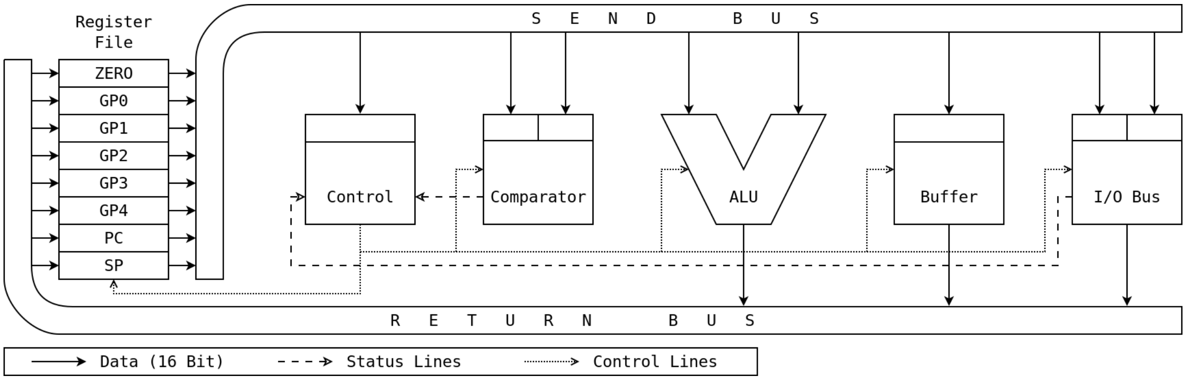
I/O The focus of this report, the input / output bus controller facilitates the sending and retrieval of data to and from many systems connected to the processor, including storage, RAM, and display drivers.

1.2 Processor Architecture

As mentioned, the processor is built around a pair of internal data buses, named SEND and RETURN. The SEND bus connects register file outputs to the inputs of all other subsystems, and the RETURN bus connects the outputs of those subsystems back to the register file inputs. A dual-bus architecture allows both buses to be used simultaneously in order to achieve rudimentary instruction pipelining. The control unit sequences parts of the processor as they perform their functions, but granular control is distributed throughout the processor. Each subsystem only needs a few signals to be provided in order to run through a complex sequence of events and perform a function.

The register file consists of one register always set to 0, five general purpose registers, one settable program counter, and a stack pointer. Whenever an operation is executed by the control unit, data will flow from one of the eight registers to one of the processor subsystem inputs over the SEND bus. Depending on the nature of the operation, data may then flow from the subsystem's output to another one of the registers over the RETURN bus. This uncomplicated paradigm makes it far simpler to envision the way data is moved and manipulated with each executed instruction.

1.3 Architectural Diagram



1.4 External Buses

The processor will be connected to every peripheral device by a pair of external, 16-bit buses, dubbed ADDRESS and DATA. The interface between the processor's internal buses and the computer's communication buses will be handled by the input / output bus controller. The controller will use device select lines to choose which peripheral to access, and an additional bus line to indicate whether data is being written or read. The ADDRESS bus will be used to indicate to devices, such as keyboards, display drivers, and memory, which address in their connected memory is being written to or read from. The DATA bus will be used by the processor to write data to devices or by devices to send data to the processor.

The bus system also offers the capability for devices to use an additonal line to issue interrupts to the processor, and to include a byte of pertinent data such as an interrupt code, indicating to the processor control unit what action to perform next. Interrupts from external devices will only be accepted when the bus controller is not in use, and they will be handled by the control unit when possible, according to priority. Yet another bus line is included to indicate, to the controller and to connected devices, when a data word is being actively asserted on the DATA bus.

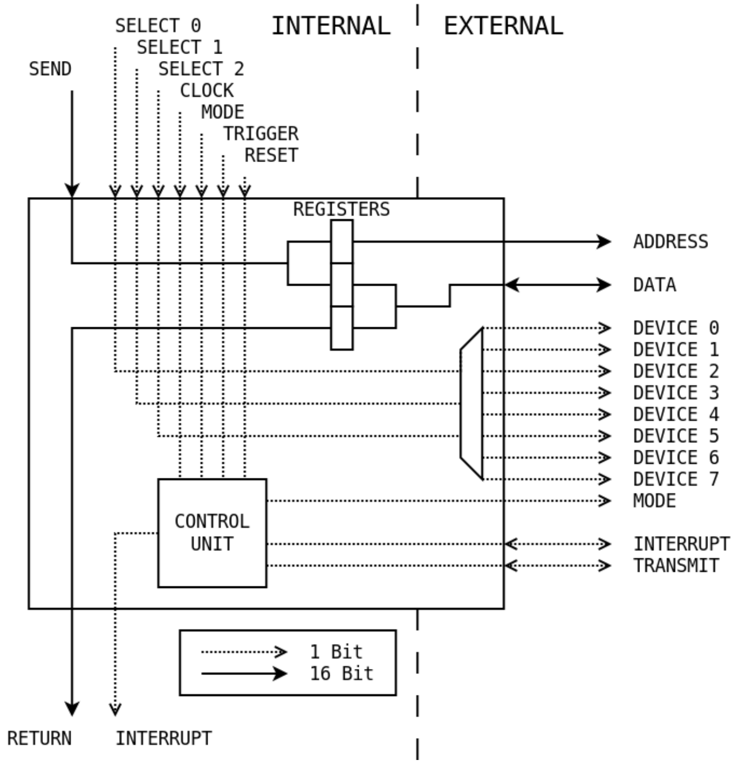
2 I / O Controller Design

2.1 Design Goals

The primary aim of this processor subsystem is to provide a high speed interface between the processor's internal SEND and RETURN buses and the ADDRESS and DATA buses of the computer as a whole. It is necessary to also provide the ability to choose a device to access, as well as whether data will be written to or read from external memory.

A secondary goal of the controller is to allow devices to send interrupt signals to the processor over the bus. Devices need to be able to signal their interrupt to the processor, and then have their request handled by the control unit at the earliest opportunity. Devices also require the ability to signal to the processor the nature of the interrupt so that the code to be executed can be determined.

2.2 General Diagram


INTERNAL
EXTERNAL
SEND
SELECT 0
SELECT 1
SELECT 2
CLOCK
MODE
TRIGGER
RESET
REGISTERS
ADDRESS
DATA
DEVICE 0
DEVICE 1
DEVICE 2
DEVICE 3
DEVICE 4
DEVICE 5
DEVICE 6
DEVICE 7
MODE
CONTROL UNIT
INTERRUPT
TRANSMIT
1 Bit
16 Bit
RETURN
INTERRUPT