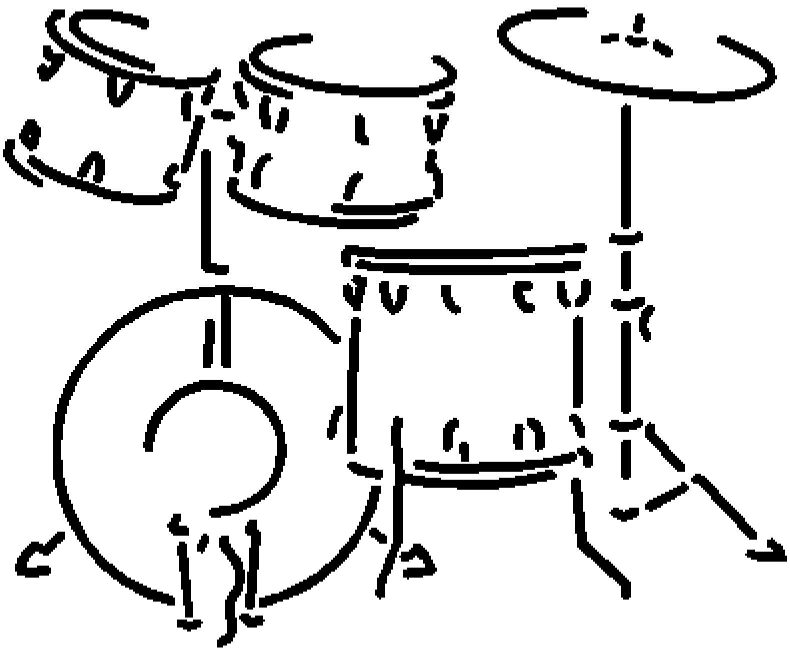
Mid Project Report

Programmable Logic Drum Machine



Student: **Gareth Morris**
Project Number: **47**
Supervisor: **Dr Seebold**
Date: **15th Jan 2002**

Abstract

A simple drum machine is to be constructed using programmable logic. The drum tones are to be generated using an integrated circuit found in early video consoles and home computers. The drum machine will be able to fully sequence one bar of rhythm, and if additional time is available extra features maybe added to the sequencer.

Contents

Index of Figures

1 Introduction

In the majority of musical styles rhythmic accompaniment is required - normally provided by a drummer. In the case where a drummer is not available, a drum kit is not practical, or in electronic music where a synthetic sound is desirable, a drum machine is often used.

A drum machine reproduces pre-programmed beats and rhythms on demand. The possibility for a mechanical device exists, but drum machines are implemented electronically in some way - with a method of programming the rhythms, and a set of drum sounds. The drum sounds maybe recorded from real life drums, or artificially produced by some kind of synthesis method. There are many possibilities for programming and, although standard techniques exist, most drum machines use their own method.

The aim of this project is to design a build a simple drum machine. The sounds will be produced by using a microchip originally designed in the 1970s for low end computer systems. This device was selected as these sounds are currently fashionable in experimental, and occasionally mainstream, music.

Although other control solutions were considered, a modern CPLD (Complex Programmable Logic Device) will be used to drive the sound chip and run the user interface. Such devices are becoming a popular choice for DSP and in new 3rd generation mobile phone systems, so the project is a real marriage of old and new technologies.

The drum machine is to sequence one full bar of 4/4 rhythm for 4 instruments, with the possibility for future expansion and live manipulation. The project will also be fully boxed and ready for creative use.

This document contains complete details of the work carried out to date – including research, planning, and initial experimentation. There is also a background section with information about some aspects unique to this project. Finally the achievements to date have been described.

The appendix also contains important information including final report outline, a technical specification, some further research and detailed time plan with commentary.

2 Background

This section describes two important pieces of background information which will helpful in understanding the rest of the project. Firstly the Yamaha YM2149 sound device to be used in the drum machine is discussed. Finally the ‘16 step’ method of drum machine programming, used in many drum machines, is explained.

2.1 Yamaha YM2149

The Yamaha YM2149 is described as being a ‘Software-Controlled Sound Generator’. It is used in some late 1980s and early 1990s home computers, games consoles, and arcade machines – such as the Atari ST. The device was attractive in these applications due to its low cost, power consumption, and simple microprocessor interface. In addition it shares its design with an earlier integrated circuit – the AY-3-8910. As these devices are no longer being manufactured it will be necessary to salvage one from redundant equipment, such as the examples shown below:

Yamaha YM2149	**General Instruments AY-3-8910**
Atari ST and derivatives	Amstrad CPC MSX1 and MSX2 Sinclair ZX Spectrum128

The differences between the circuits are small – the Yamaha YM2149 has a few additional features, such as the support of higher clock rates (with a selectable frequency divider) and slightly improved performance. It will doubtless be easier to find AY-3-8910 devices as they are used more frequently and in equipment with a lower second hand value.

Both devices have sufficient square wave, noise and envelope generators to produce three independent channels of audio, although the noise frequency must be the same for each channel. All these functions are configured by writing to the device’s internal registers. Sound generation is digital, but the output passes through a 5 bit D/A converter, which uses a non-linear quantiser, to produce three 1v p-p analogue outputs.

An additional feature of these devices is two bi-directional 8bit wide parallel buffers. They are frequently put to use in low-cost home computers for such tasks as I/O ports. This facility will be ignored in this project, except for producing simple test outputs.

The packaging of both microchips is a 40-pin DIL case. All pins (except for the analogue output) are TTL compatible. The power supply is a standard 5 volts. Therefore there should be no difficult to solve level compatibility problems.

The device was primarily selected as it is difficult to program using pure logic, making for a more challenging project. A secondary consideration is the nostalgic timbre of this device, recalling memories of a past era - for this reason many experimental musicians, such as LotekStyle, Gwem and the Gwemettes, and Bodenstandig 2000, are currently using these types of sounds.

2.2 16 Step Drum Sequencer

The project is to use a method found in many drum machines. Each bar of rhythm is divided up into 16 steps. When the rhythm is played back each of these 16 steps are cycled through in sequence. When the sixteenth step is reached playback returns to the first.

In order to show if a drum is to sound at the start of a step LEDs are illuminated on the front panel of the drum machine, this is shown below, with a comparative musical representation:

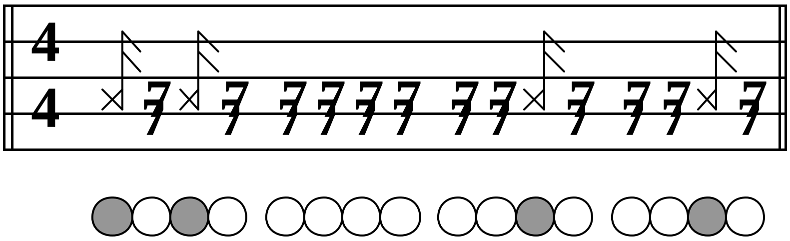


Figure 1: Drum to sound on the first, third, eleventh, and fifteenth steps of this bar

These LEDs are put into groups of four in order to show where the beats would fall in 4/4 time.

A user may adjust which steps the drum 'hits' occur by pressing buttons positioned below each of the LEDs. Pressing one of these buttons toggles whether or not a drum sound occurs when playback reaches this step.

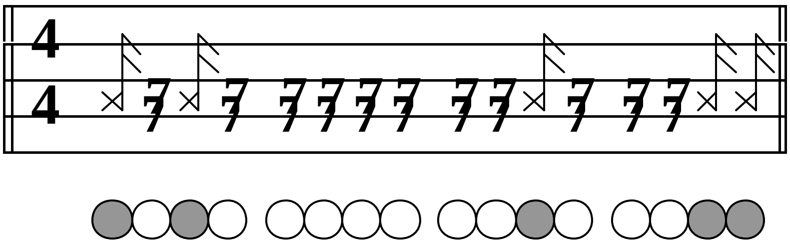


Figure 2: The sixteenth button is pressed – this toggles the status of the 16^{th} step so a drum will now sound

Each of the different drum sounds can be selected and programmed in this way. If a drum sound has not completed by the time it is sounded again (eg between beats fifteen and sixteen in the diagram above) then the remainder of the sound is cut short. This emulates what happens on a real drum kit.

For further realism the open hi-hat sound is cut short by the next closed hi-hat sound. In addition closed and open hi-hat sounds cannot occur together – priority is given to the closed hi-hat.

3 Research

A period of research was carried out at the start of the project. The aim was to ascertain the best method of controlling the sound chip. The results of this research have been summarised in this section. Furthermore there is additional research in the appendix comparing my project with similar products.

3.1 Solutions

Although the programmable logic solution was selected early on the project, different possibilities for the drum machine were considered.

Microprocessor (eg 68???, Z80)
Microprocessors, of course, do not possess the interface possibilities of a microcontroller and require additional hardware in order to perform useful tasks. The advantage of this is that microprocessor can be more optimised to computation, and the external hardware required can be customised to the particular application. Microprocessors are inherently concurrent, a disadvantage in a system where inputs should ideally be processed in parallel, and an interrupt controller will need to be designed. The requirement for an external interface is an advantage in that the hardware may be designed for the minimum functionality required (for instance reducing cost). However in the short time available to build the drum machine this may cause a problem. It is likely that in the small system being built that the power of a microprocessor would not fully be utilised.

Microcontroller (eg PIC, 8051)
Microcontrollers, are typically less powerful than microprocessor, but have many on chip peripherals and features – such as memory, serial ports etc. This makes interfacing to external devices easier, and is often only a matter of simple logic, or clever I/O coding in the microcontroller itself. A device of this sort would be a good solution to use for the drum machine, and there are many development tools available. Due to their popularity these devices are often obtained at low cost.

Discrete Logic
Although drum machines have been built using discrete logic, these devices are low density(meaning a large and bulky end result) and the circuits are less easy to reconfigure in order to fix problems. Discrete logic would not be an ideal solution.

Programmable Logic
Programmable logic is increasing in popularity with easy to use low cost devices from companies such as Altera and Xilinx. Programmable logic devices may not be so well supported, as for example a PIC microcontroller, there are sufficient tools available. External hardware maybe made very simply, as all digital logic needed can be programmed into the chip. Obviously these devices are inherently parallel, and would be more efficient in such a system. Some style of programmable logic would be a good choice for the drum machine.

The choice between microcontroller and programmable logic was finally decided by personal reasons, the author having extensive experience with Xilinx programmable logic devices. The actual device series selected was the XC95?? CPLD because of its non-volatility.

4 Implementation

4.1 System level block diagram

The diagram below shows a broad overview of the drum machine system components, and how they interact. The generic term ‘Programmable Logic’ is used as the architecture the final system will be based on has not yet been decided. The major part of the work lies in this programmable logic section, with the audio and interface components being relatively simple to implement.

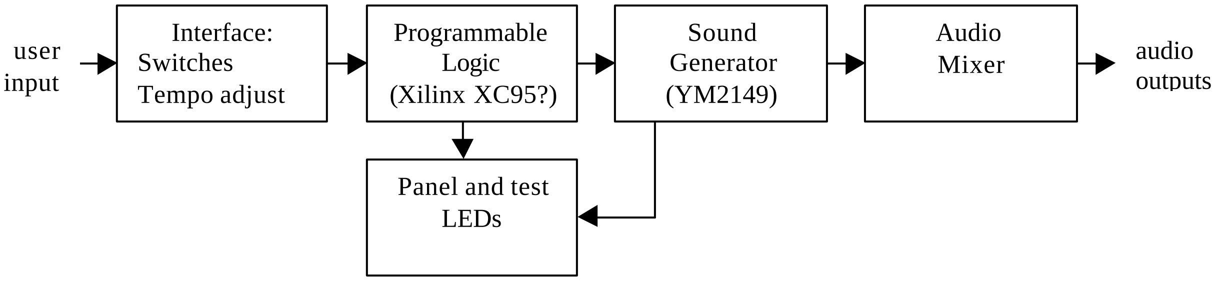


Figure 3: Proposed system level block diagram of completed system

Explanation

- The user configures the interface of the drum machine: including tempo and rhythm for each instrument.
- The CPLD senses these changes and makes adjustments to itself and the panel LEDs.
- When the user initiates play the CPLD triggers the sound chip to produce sounds when required.
- The output of the sound chip passes through an audio-type mixer for user adjustment of each instrument’s output level.
- The electrical audio output signal may be routed for external amplification etc.

Programmable logic description
Broadly speaking, the design of the logic which forms the heart of the drum machine system is simple. Two large function blocks control the sequencing and driving of the YM sound chip. The interface between these two modules is a series of parallel pulses which occur when an instrument is to sound.

Much of the interface to the sequencer block is undefined – however the sequencer is likely to be the simplest design problem to solve, and will probably use some kind of simple counter/multiplexer arrangement.

The driving logic for the YM sound chip is a great deal more complex and for this reason a special test circuit is to be built in order to develop this part. This is described in the following section.

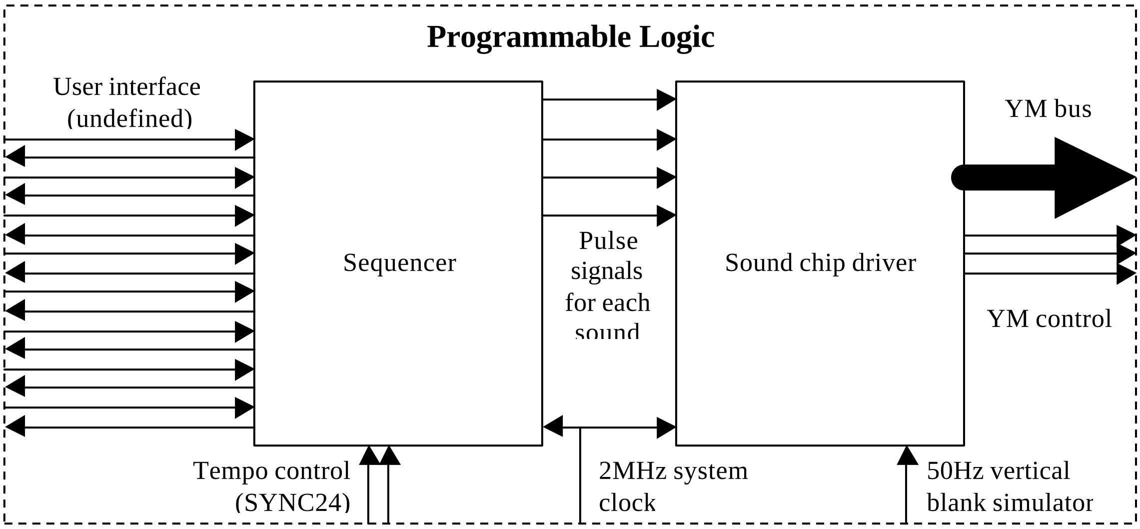


Figure 4: Block diagram of final logic design

4.2 Initial test circuit

The design of the drum machine's sequencer would seem to be relatively straight forward, with most of the risk lying in the tone generation section. Therefore a test board is to be made, containing a prototype version of the YM sound chip circuit and driver, complete with associated mixing and amplification.

Individual drum sounds are to be triggered manually by use of switches. A full simulation of this part of the design is not possible, as the sounds will need to be configured partly by ear.

In addition to finalising the tone generation circuitry, the test board will help to determine the size and type of the programmable logic device required in the final design.

A block diagram, showing the proposed layout of this test board is shown below:

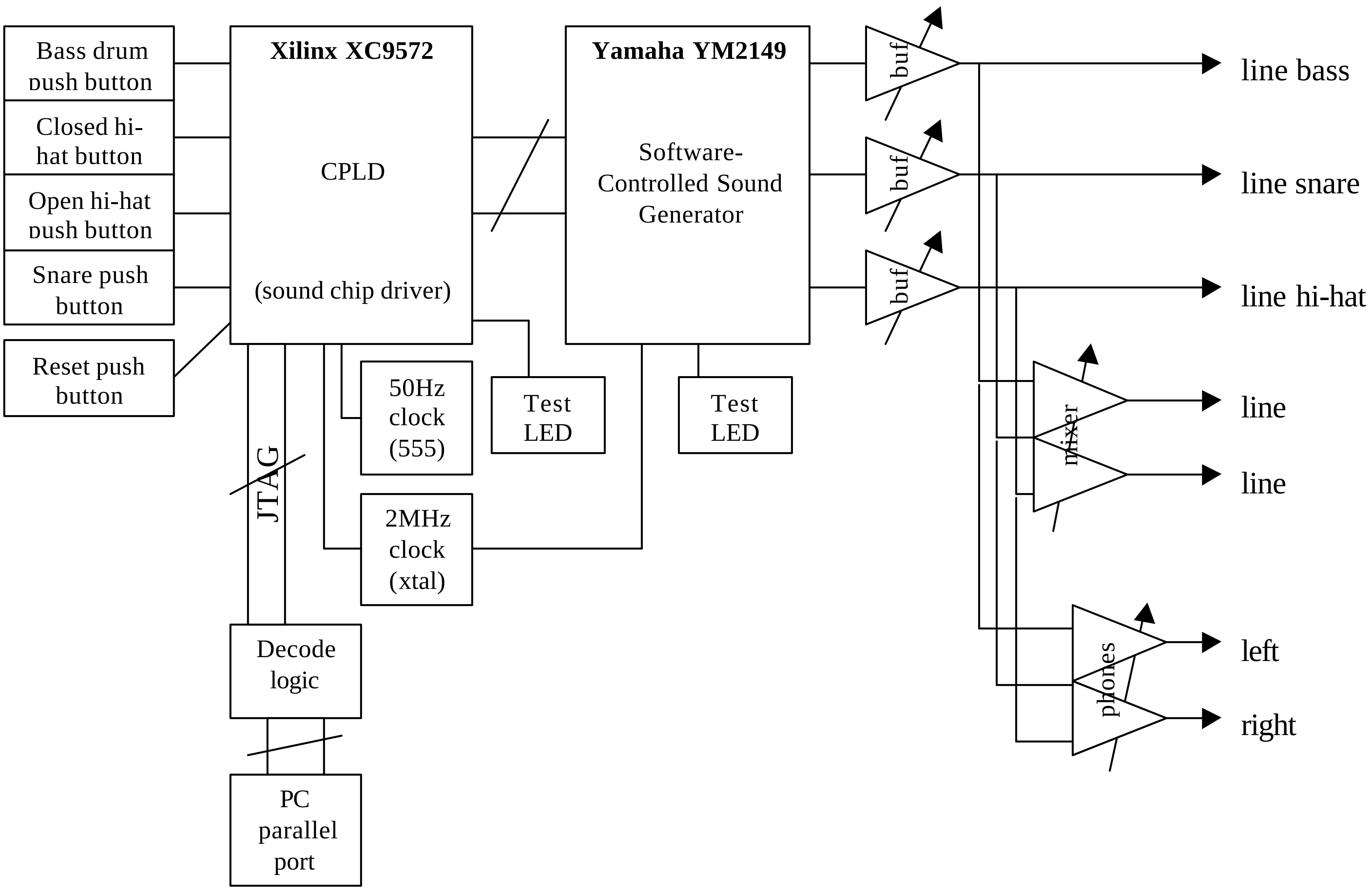


Figure 5: Block diagram of tone generator test circuit

4.3 Points of Note

The tester triggers each sound by pushing buttons. An additional button exists for performing a reset of the sound chip.
The function of the two test LEDs is to a visual indication of the functioning of any part of the logic. They can be ‘rewired’ internally to the CPLD.
The JTAG interface shown allows the in-system programming of the CPLD, so that the device may be easily reconfigured to fix bugs.
Closed and open hi-hat sounds are to be combined on one channel of the YM2149, with priority given to the closed hi-hat sound.
The 50Hz clock simulates the 50Hz vertical blanking signal used by video games consoles. Changes to the drum tones will only occur at these intervals, for authentic sound quality. As the sounds will only last a few of these periods, very precise timing of this clock is not required.
The YM2149 is clocked at 2MHz allowing for minimum and maximum frequencies of around 30Hz and 125kHz. This is the same as Atari ST home computer. For convenience the CPLD also uses this clock, although the interface between the two devices is entirely asynchronous.
The audio section represented to the right of the diagram is to comprise of low noise op-amps and a stereo headphone amplifier. Exact functionality is described in the technical specification – Appendix C -

4.4 Sound chip driver

In the final design the logic for the sequencer and the sound chip driver will be designed as separate modules in the same logic device. This driver is also required for the test circuit described in the previous section. Therefore a lower level description of this is appropriate and a block diagram with explanations is shown below:

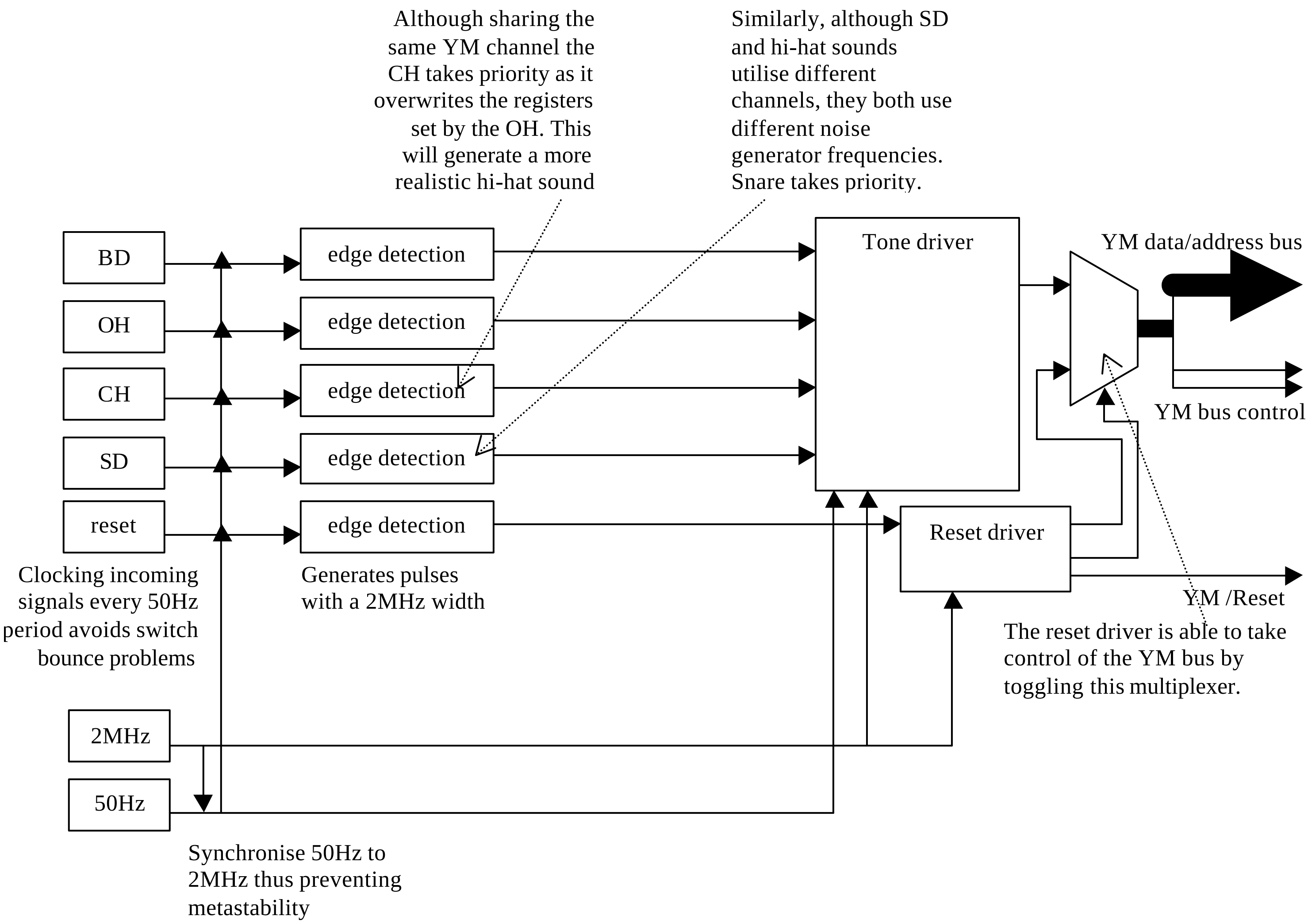


Figure 6: Block diagram of sound chip driver

The reset and sound generators are made separate as the sound generator is required to be triggered at 50Hz intervals in order to vary the output. The reset, in contrast, does not need this – but instead writes to a whole range of different registers.

The reset generator takes to form of a simple counter, whose output is decoded to form the data and address bytes, and also the BC1/BDIR bus control signals.

5 Progress to date

So far the project has advanced well, albeit with the design work slightly behind schedule. The test board discussed in section 4.2 has been fully constructed and almost completely debugged. This proves the principal behind the project, and shows that it should be possible.

All the sounds are working and can be simply triggered by using a push button. The audio section is working with no problems. The test board is not quite completed as the sounds themselves need some adjustment to improve their tonal quality.

There were some problems encountered in achieving this, which will be fully documented in the final project report.

The following table shows the current status of each of the project time plan tasks for this semester (see Appendix D -)

Task	**Status**
Setup project with supervisor	Completed
Research	Completed
Rough specification	Completed
First time plan	Completed
Second time plan	Completed
Design initial test CPLD	Completed
Design initial test circuit	Completed
Design test circuit PCB	Completed
Order components for test circuit	Completed
Build test circuit	Completed
Debug test circuit	Almost completed
Detailed specification	Completed
Third time plan	Completed
Design final circuit	Begun
Design final PCB	Not started
Revision/exams	Completed
Mid project report	Completed

As can been seen about 90% of work time-tabled has been carried out. There is sufficient contingency in the spring semester time plan to recover this 10%. For instance, currently no more than 2 weeks is anticipated for debugging the final circuit and bar sequencer, and certainly less than the 6 weeks timetabled. Therefore the progress to date has been good, and the project should be completed. In solving the various problems encountered, much has already been learnt.

6 References

Document	Description	Publisher	Date/Version
YM2149 Software Controlled Sound Generator	data sheet	Yamaha Corp.	1987
AY-3-8910 Programmable Sound Generator	data manual	General Instrument Corp.	1979
Atari ST Computer System	‘Engineering Hardware Specification’	Atari Corp.	07 Jan 1986
XC9500 ISP CPLD Family	data sheet	Xilinx Corp.	12 Sep 1999

7 Glossary

Atari ST	16bit home computer made by the Atari Corporation from 1985
AY-3-8910	The General Instrument’s device the YM2149 copies. Used in Spectrum and Amstrad home computers among others
BPM	Beats Per Minute, a measure of the tempo of a piece of music
CPLD	Complex Programmable Logic Device, a type of programmable logic architecture – typically non-volatile
Drum machine	An electronic musical instrument which may be programmed to reproduce rhythms
ISP	In System Programmable. Describes a device which maybe programmed in the field without removal from its socket.
MIDI	Musical Instrument Digital Interface, a serial data protocol which may be used to interface between different instruments using the standard
Programmable logic	Digital logic (which may include gates and/or storage elements) which may be configured by the designer
Sequencer	In the context of a drum machine the sequencer refers to the part of the digital logic and interface which controls the timing of the sounds
SYNC24	Antiquated system for synchronising drum machines
Tempo	The rate at which music occurs. Measured in beats per minute (bpm).
Timbre	Refers to the perceived psychological sound characteristics
Vertical blanking	Old video games systems use an interlaced display system, which outputs alternate lines at 50Hz periods. The electronics of the display system generates an interrupt to the processor at these intervals. Many sound and other functions also use this signal as a real time timing reference.
Xilinx	Manufacturer of programmable logic devices.
YM2149	Sound generation device made by the Yamaha corporation. Copies the AY-3-8910 and adds some minor enhancements.

8 Appendix

Appendix A - Final Report Outline

The final report for this project will comprise of the following sections:

1. Executive summary (what has been achieved)
2. Contents etc….
3. Introduction (becoming progressively more detailed, with some background material)
4. Implementation (a chronological summary)
5. Implementation (description of the final design)
6. Possible improvements
7. Conclusion
8. Glossary etc…
9. Appendix

Appendix B - Competing products

Both obsolete products and equipment in production were compared to my project.

Roland - TR series
The drum machines in the TR (Transistor Rhythm) series were manufactured by the Roland Musical Instrument Corporation during the 1980s. All of the units, bar one, use the 16 step interface described in section 2.2. However these instruments are now not only famous for this step editor, but also for their high quality sounds – which used samples and analogue circuitry. Due to this many of the models now command high prices on the second hand market.

One of the first models (1983) in this series was the TR-606. It is this product which will be used as a comparative model for developing the project.

Korg - Electribe R
This drum machine is in production and, like many, copies the interface of the Roland TR series – but offers many enhancements. The sound generator uses DSP technology in order to reproduce the old fashioned sounds made by analogue circuits. Enhancements to the interface include features to improve usage in a live situation. There is the possibility that some of these extra features can be included in to the project at a later stage.

Elektron - SID Station
Although not a drum machine, this sound generator uses the sound chip from the Commodore 64 home computer, and adds a highly developed MIDI interface and control system. This marks a growing trend in dance music production – and sounds from the C64 have been used in several chart hits. This sound chip is different, and more advanced, than the one proposed for the drum machine.

Propellerheads - Rebirth
Rebirth is computer software available for modern day PCs that emulates drum machines from the Roland TR series by using samples and software filtering. This software is popular due to its low cost compared to ‘real’ second-hand equipment, and the fact the sound is almost indistinguishable from the originals. It is possible to combine this software with other music packages into a so-called ‘virtual studio’ all running inside a PC.

Appendix C - Technical Specification

Sequencer		
	Device:	Xilinx XC9572 CPLD
	Resolution:	1 bar, 16 steps
	Tempo:	Continuously adjustable 30-300bpm
	External i/f:	SYNC24 input/output for synchronisation
Tone generator		
	Device:	Yamaha YM2149 “Software Controlled Sound Generator”
	Presets:	Bass and Snare drums. Open and closed hi-hats. Hi-hat sounds not available simultaneously. Sounds may not be used adjusted.
Audio		
	Outputs:	3x ¼” mono jack outputs (one for each sound type). 1v p-p, 1k impedance 3 channel mixer with user adjustable levels 2x ¼” mono jack outputs for mixed signal. 2v p-p (adjustable), 1k impedance ¼” stereo jack output for headphones. 1W RMS (adjustable), 16 ohm impedance. Mute switches for each sound.
Power		
	Voltage:	9v DC input, rectified to allow inputs of unspecified polarity
	Current usage:	300mA

Appendix D - Time Plan

Tasks

One may also refer to the time plan gantt chart for the position of these tasks over the course of the project.

Research

In order to ascertain what design solutions would be implemented, a period of research was required. This took the form of obtaining manufactures data sheets, and examining previous implementations. The results are summarised in sections 2 and 3 of this report.

Specification

Based on the research a broad technical specification of the project could be drawn up, in order to describe what would be implemented. This is reproduced in appendix (Appendix C -.

Initial test CPLD

As described in section 4 the most risk in the project is driving the YM2149 sound chip from the logic. A test circuit will be built to finalise this. In order to design this test circuit the type of CPLD device must be known – to find this the CPLD must first be designed. The high level implementation of this is defined in section 4.4.

Initial test circuit plus PCB
Once the CPLD has been designed work may proceed with designing and building the test circuit. The PCB for this circuit will be strip or veroboard – which is easy to modify, yet more reliable and permanent than plug-in breadboard prototyping.

Final Circuit plus PCB
Using the debugged test circuit as a basis the final drum machine schematic maybe designed, including the 16 step sequencer. A custom PCB will be designed and etched for this – to improve performance and make construction easier.

Final CPLD
Once the final circuit is fixed then work can begin on the programmable logic for it. Parts of the test circuit CPLD can be reused, along with some extra logic.

Box
In order to contain the instrument in a portable and useful way a box will be constructed. This will probably be in the form of a modified commercially available container – of plastic or metal. The box will also include a battery container for full portable use, and some kind of front panel artwork.

Testing
Although testing of the individual modules will occur throughout the course of the project, some time at the end is reserved for testing. Some pre-defined test bars will be programmed into the unit and their execution tested. These bars will test extremes of usage, and also special features of the sequencer (such as non-simultaneous hi-hat effects) – they will be fully documented in the final report. In order to test the audio section a spectrum analyser will be used to measure power, frequency and noise figures to compare to the specifications.

Time plan commentary
The first seven to eight weeks are concerned with the administration of setting up the project and developing the initial test circuit described in section 4.2. These tasks inevitably come at the start of the project, where a complete solution is not fully defined. The remainder of the time until the Christmas break is taken up with designing the completed circuit based on the findings of the initial test circuit.

During the spring semester the final circuit will be constructed and debugged, and made ready for the final presentation.

Contingencies
The time plan makes the assumption that there will be no problems throughout the project. As this is unlikely however contingencies are possible:

- If there are big problems simply getting test circuit working is a possibility
- A raw circuit board could be presented rather than a boxed product
- The entire design could be produced on veroboard, rather than a special PCB
- Parts of the logic could be simulated rather than implemented if there is insufficient space on the CPLD
- One week exists for work at the end of the spring semester, and the possibility for working over the Easter break has been over looked.
- Although all four sounds could be operational, the sequencer could just work for one of them in case of problems.

Gantt Chart

Winter Semester Task	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	Xmas
Setup project with supervisor	■															
Research	■															
Rough specification	■	■														
First time plan	■															
Second time plan		■														
Design initial test CPLD		■	■	■												
Design initial test circuit					■											
Design test circuit PCB				■	■											
Order components for test circuit						■										
Build test circuit							■	■								
Debug test circuit								■	■							
Detailed specification										■	■					
Third time plan											■					
Design final circuit											■	■				
Design final PCB																■
Revision/exams													■	■	■	
Mid project report																■
Spring Semester Task																
Fourth time plan	■															
Order components for final circuit	■															
Build final circuit		■														
Debug final circuit			■	■	■											
Design & debug PLD bar sequencer						■	■	■								
Build box									■	■						
Testing											■	■				
Final time plan													■			
Final report													■	■	■	

This page is left blank