

SIGNAL SWITCHING MATRICES FOR REMOTE LABORATORIES - EXISTING SOLUTIONS AND A NOVEL APPROACH USING FPGA TECHNOLOGY

ZENZEROVIC, P[aolo]; MACUKA, D[aniel] & KRISKOVIC, D[omagoj]

Abstract: *The concept of software controlled flexible hardware interconnections in the field of remote laboratory systems is introduced in this paper. Existing solutions for this problem are reviewed and a novel approach using FPGA technology is proposed. The FPGA implementation method is described in details and compared to the existing solutions in terms of implementation costs, surface requirements, universality, and feasibility. It is shown that the novel approach using FPGA technology for the implementation of software controlled hardware interconnections stands as the most desirable solution for this purpose.*

Keywords: *remote laboratory, e-learning, engineering education, switching matrix, FPGA*

1. INTRODUCTION

With today's ever growing and expanding industry there is an ever increasing need for electrical engineers, as well as engineers in all other fields of industry. High schools and universities tend to adapt to that need by increasing the number of places available for study in these areas. This increasing number of places often results in decreased quality of the educational process. The part of the educational process that is most affected by this increasing number of students is the transfer of practical knowledge from the professor to students because of the lack of adequate laboratory spaces and equipment [1].

One possible solution for this problem is the usage of remote laboratories. Remote laboratories are specialized laboratories which can be accessed by a network infrastructure, most commonly through Internet [2]. The basic idea behind this concept is to enable access to the laboratory equipment by using a personal computer with the effect that closely resembles to being physically present in the laboratory [3].

The existing remote laboratories mainly consist of the main microcontroller unit, in which the user code is downloaded, the peripherals with which the user remotely interacts through the mentioned microcontroller unit and the most important thing, the flexible wiring module that is used for redirecting the user signals from any/all microcontroller pins to any/all peripherals pins.

Various switches can be used to enable this redirecting. Some existing solutions for the flexible wiring module as well as the new approach to this problem will be presented and compared in this paper.

2. EXISTING SOLUTIONS FOR FLEXIBLE HARDWARE INTERCONNECTIONS

2.1 Flexible interconnections in remote laboratory systems

In remote laboratory systems, flexible hardware interconnecting or flexible wiring is used to enable the redirecting of user signals between the main microcontroller unit, which is the central part of a remote laboratory system, and its peripherals [4]. Different switching components can be used to redirect these signals. These components can be divided into two main groups, those that are controlled manually (e.g. toggle switches) and those that are software controlled (e.g. different types of relays, multiplexers/demultiplexers etc.).

The existing flexible interconnection solutions, using both manually and software controlled switching components, with the assumption that forty inputs and forty outputs are needed, will be presented in the following text.

2.2 Toggle switches

The basic solution that arises for the realization of hardware interconnecting is the toggle switch. This is a manually controlled switch that has two states, on and off.

To enable the connection of any/all inputs to any/all outputs, one thousand and six hundred of these switches are needed. The implementation costs, as well as the amount of volume and surface needed when using toggle switches, are shown in Table 1.

Toggle switch implementation characteristics		
Quantity	pieces	1600
Total cost	USD	1040
Total surface	cm ²	3500
Total volume	cm ³	11200

Tab. 1. Toggle switch implementation characteristics

2.3 Mechanical relays

Flexible interconnection can also be implemented by using relays. Relays are electrically operated switches. Considering their switching principles, there are two main types of relays, those that have moving parts for switching, and those that have no moving parts but they use semiconductor devices for switching instead.

In this case flexible interconnecting is implemented using mechanical relays. These relays have two overlapping iron nodes that are separated in non-

operating conditions and only touch each other when the external voltage is applied.

One thousand and six hundred of these relays are needed to enable the interconnection of any/all inputs to any/all outputs. The implementation costs, as well as the amount of volume and surface needed when using reed relays, are shown in Table 2.

Mechanical relay implementation characteristics		
Quantity	pieces	1600
Total cost	USD	2225
Total surface	cm ²	5568
Total volume	cm ³	8185

Tab. 2. Mechanical relay implementation characteristics

2.4 Solid state relays

Solid state relays, compared to mechanical relays, have no moving parts. Instead, they use semiconductor devices for switching.

Like both toggle switches and mechanical relays, one thousand and six-hundred solid state relays are needed to implement a flexible interconnection of any/all inputs to any/all outputs. The implementation costs, as well as the amount of volume and surface needed when using solid state relays, are shown in Table 3.

Solid state relay implementation characteristics		
Quantity	pieces	1600
Total cost	USD	6320
Total surface	cm ²	5557
Total volume	cm ³	13892

Tab. 3. Solid state relay implementation characteristics

2.5 Digital multiplexer and demultiplexer switches

Digital multiplexers and demultiplexers offer another solution for the implementation of flexible interconnection module. These devices allow the data flow in one direction only and the flexible interconnection module must have the bidirectional property. That is why both multiplexer and demultiplexer components have to be used in this implementation.

Multiplexers provide the possibility to connect one of multiple inputs to one output, and the demultiplexer provides the possibility to connect one of multiple outputs that in this case serve as inputs, to one of the inputs, that in this case serve as outputs. In this way the bidirectional property of flexible interconnection is achieved. The results for this category are shown in table 4.

Digital mux/demux implementation characteristics		
Quantity	pieces	240+240
Total cost	USD	360
Total surface	cm ²	1302

Tab. 4. Digital multiplexer and demultiplexer implementation characteristics

2.6 Analog multiplexer/demultiplexer switches

Another existing solution for the implementation of flexible interconnection is the analog multiplexer/demultiplexer component. This component has bidirectional pins so it can work as a multiplexer, allowing the user to address one of multiple inputs to one

output, or as a demultiplexer, allowing the user to address an input to one of multiple outputs. There are various sizes of these components considering the number of inputs/outputs.

To allow the flexible interconnection of any/all inputs to any/all outputs, two hundred and forty “8 to 1” multiplexer/demultiplexer components are used. The implementation costs, as well as the amount of surface needed when using multiplexer/demultiplexer, are shown in Table 5.

Analog mux/demux switch implementation characteristics		
Quantity	pieces	240
Total cost	USD	343
Total surface	cm ²	262

Tab. 5. Analog multiplexer and demultiplexer implementation characteristics

2.7 Crosspoint switches

The last of the existing solutions for the implementation of flexible interconnection shown in this paper is the crosspoint switch.

This is a type of switch that can connect multiple inputs to multiple outputs in a matrix manner, using an assembly of individual switches between these inputs and outputs. This switch has the bidirectional property, so it allows the flow of data in both directions.

Like multiplexer/demultiplexer components, there are various sizes of these switches considering the number of inputs/outputs. For the purpose of this solution, twenty five “8 to 8” crosspoint switches are used to allow the flexible interconnection of any/all inputs to any/all outputs. The implementation costs, as well as the amount of surface needed when using crosspoint switches, are shown in Table 6.

Crosspoint switch implementation characteristics		
Quantity	pieces	25
Total cost	USD	236
Total surface	cm ²	25

Tab. 6. Crosspoint implementation characteristics

3. A NOVEL APPROACH USING FPGA TECHNOLOGY

3.1 Implementation method

A field-programmable gate array (FPGA) is an integrated circuit designed to be configured by a customer or a designer after manufacturing and that is why it is called field-programmable. FPGAs contain programmable logic components called “logic blocks”, and a hierarchy of reconfigurable interconnections that allow the blocks to be “wired together”, somewhat like many changeable logic gates that can be wired together into many different configurations. That enables the user to set up their own hardware structure inside a single FPGA device. With this property it is possible to implement a flexible interconnection module inside a single FPGA device.

The schematic of the implementation is shown in Figure 1. The simplest way to describe the implementation method is to start from switches that are lowest in the implementation hierarchy. In Figure 1, their

names are “Switch 1” to “Switch 40”. These are bidirectional switches, implemented by using three tristate buffers. Two of these buffers are used to determine the direction of data flow, which is done by setting the correct “Switch control” pin, shown in Figure 1, to the appropriate state. The third buffer is used to enable the data flow, thus controlling which switch is active at the given moment. This buffer is controlled by the appropriate “Switch enable” pin. Each of these switches has one data input pin separated from other switches, and one output pin that is common to all switches. Forty of these switches form one “40-to-1” switch that can also be seen in Figure 1. This switch has forty inputs and one output and, since forty outputs are needed, forty “40-to-1” switches are used. Inputs of all these switches are connected to a common bus that is connected to user input pins. Outputs of these switches are connected separately to user output pins as shown in Figure 1. Each one of these switches has an address input pin that allows the user to control which switch is active at a given time. In this way the user can choose which one of the input pins is connected to which output pin and vice versa.

3.2 Implementation results

By the above mentioned method flexible interconnection of forty to forty pins was successfully implemented inside a single FPGA device. It is worth mentioning that, although input and output pins are bidirectional, the user has to know the direction of data flow in advance and set the appropriate “Control” pin to allow the flow of data in the desired direction.

For the purpose of this implementation 70% of the given FPGAs capacity was used. Implementation costs, as well as the amount of surface needed when using this FPGA for the implementation of flexible interconnection are shown in Table 7.

FPGA switch implementation characteristics		
Quantity	pieces	1
Total cost	USD	164
Total surface	cm ²	16

Tab. 7. FPGA switch implementation characteristics

4. COMPARISON OF EXISTING FLEXIBLE HARDWARE INTERCONNECTION SOLUTIONS AND A NEW SOLUTION USING FPGA TECHNOLOGY

The first of the proposed existing solutions for the flexible hardware interconnection, the toggle switch, is a solution that has the advantage in the simplicity of implementation and its bidirectional property so the user doesn't have to know the direction of data flow in advance. One other advantage of this solution is the possibility to interconnect both digital and analog signals. Beside these advantages, this implementation solution has quite a few disadvantages. One thousand and six hundred of these switches are needed for the implementation purposes which result in a large surface requirement, as shown in Figure 3. When implementation price is considered, these switches have an advantage over the solutions using relays, as it can be seen in Figure 2, but still the expenses are considerably higher as

compared to digital mux and demux, analog mux/demux, crosspoint and FPGA solutions. The biggest disadvantage of this solution is the lack of the software control possibility and the interconnection can only be done manually, so this implementation is not suitable for the remote laboratory hardware interconnection purposes.

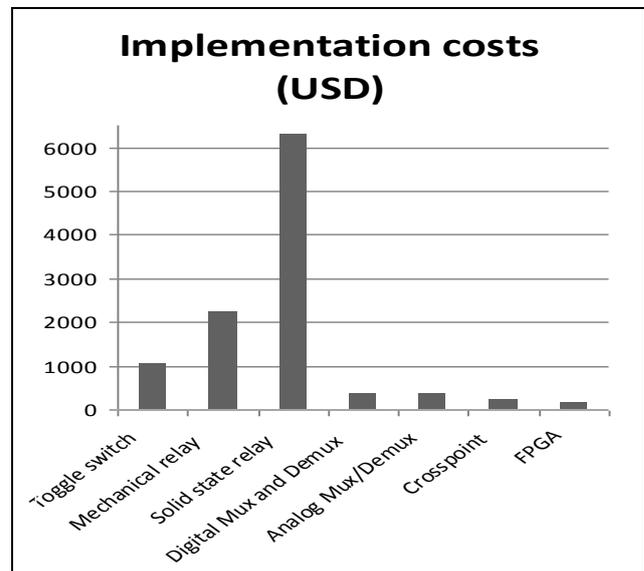


Fig. 2. Flexible interconnection implementation costs for various solutions

Reed and solid state relays are the two other existing solutions that were proposed. Like toggle switch, both of these solutions have the bidirectional property so the user doesn't have to know the data flow direction in advance and they too have the ability to interconnect both digital and analog signals. Relays have a big advantage over toggle switches as they can be software controlled. Implementation costs and surface requirements are two big disadvantages of both of these solutions. In both solutions, one thousand and six hundred relays are needed, which results in considerably higher implementation costs than other solutions, as shown in Figure 2. As compared to other solutions, surface requirements for both relay implementations are much higher, as shown in Figure 3.

Another proposed solution, using digital multiplexer and demultiplexer devices, has an advantage when implementation costs are considered, as shown in Figure 2. Like relay solutions, it can also be software controlled but the user has to know in advance whether to use multiplexers or demultiplexers depending on the direction of data flow, and it has only digital inputs and outputs.

This solution requires fewer components than the toggle switch and relay solutions, but still has a considerably larger surface requirement than the analog mux/demux, crosspoint and FPGA solutions, as seen in Figure 3.

The solution which uses analog mux/demux components is very similar to the previous solution, when it comes to implementation costs shown in Figure 2, but it requires two times less components and is bidirectional so the user doesn't have to know the direction of data flow in advance. When the surface requirement is considered, this solution has a much smaller surface requirement than the already compared existing solutions, but that is still considerably higher as compared to the crosspoint and FPGA solutions.

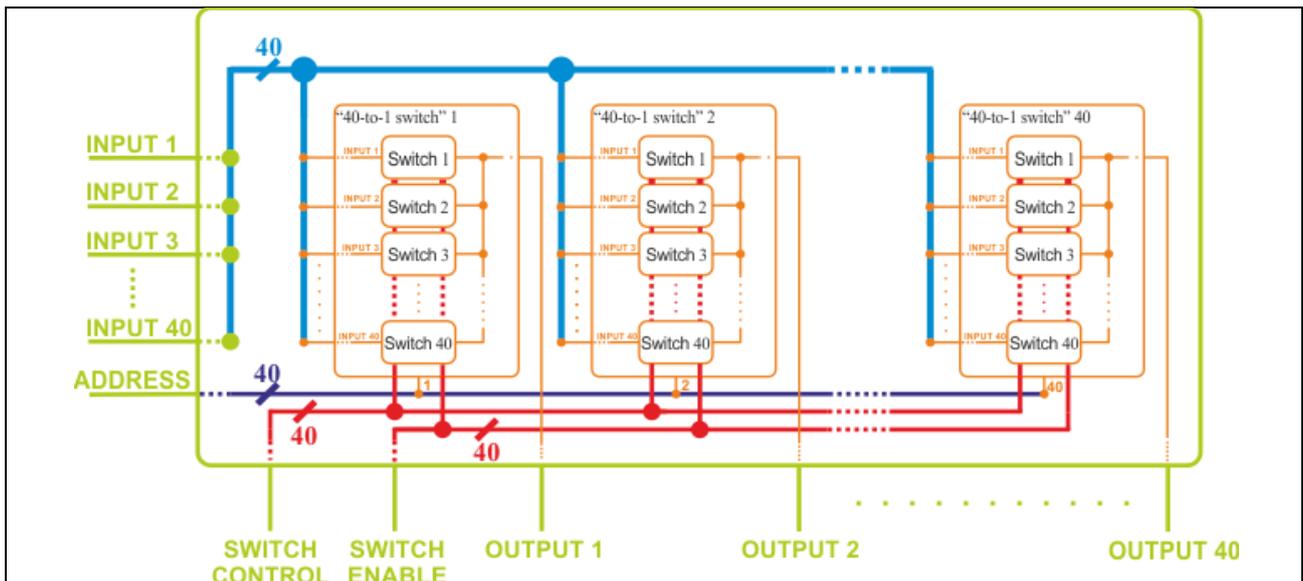


Fig. 1. Flexible interconnection implementation using FPGA

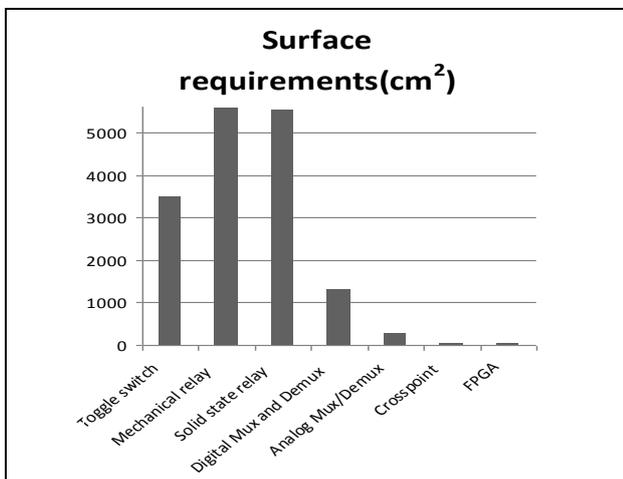


Fig. 3. Surface requirements for various flexible interconnection implementation solution

The crosspoint solution has the lowest implementation costs and surface requirements of all solutions except the FPGA solution, and this can be seen respectively in Figure 2 and Figure 3. The advantage over the FPGA solution is the bidirectional property of this switch, but on the other side, it can only interconnect analog signals.

As mentioned, the FPGA device has the lowest implementation costs and surface requirements of all the proposed solutions and requires far less components than all other solutions, in fact only one FPGA device is needed to implement the flexible hardware interconnection. The only disadvantage of the FPGA solution is that it can interconnect only digital signals, so the user has to know the direction of data flow in advance.

5. CONCLUSION

The software controlled flexible hardware interconnection solution that is implemented using FPGA devices has an insensibly smaller surface than the crosspoint switch solution, but significantly smaller than all the other solutions, as seen in Figure 3.

When it comes to implementation costs, the FPGA solution costs are lower than all the other implementation solutions. The only limitation of this solution is that it can interconnect only digital signals and the user has to know the direction of data flow in advance. This presents a problem when using peripherals that have various pins with different data flow directions. This limitation, however, has been easily solved, as described in Section 3.1.

Considering the above facts, the novel approach using FPGA technology regarding the implementation of software controlled hardware interconnections stands out as the most desirable solution for this purpose.

6. ACKNOWLEDGEMENTS

The authors appreciate the support of the Faculty of Electrical Engineering of the University of Rijeka, the CEEPUS (CIII-HR-0108) network and the Croatian Society for Educational technology in this project.

7. REFERENCES

- [1] Barr and J. Tagg, „From teachnig to learning: A new paradigm in undergraduate education“, Hildref Publications, 27 (1995), 6, 198-2000
- [2] Sousa, N., Alves, G. R., Gericota, M. G.: „An integrated reusable remote laboratory to complement electronics teaching“, IEEE Transactions on Learning Technology, 3 (2010), 3, 265-271
- [3] Zenzerovic, P., Macuka, D., Kriskovic, D.: „Remote Laboratory for Automation System Design“, *Annals of DAAAM for 2011 & Proceedings of the 22nd International DAAAM Symposium*, Vienna, Austria 2011, 1655-1656
- [4] Zenzerović, P.: *Remote laboratory for the desing of microcontroller systems: prototype solution and design of development platform*, Faculty of Engineering, University of Rijeka, Rijeka, 2010., (in Croatian)