FUSE

Best Practice Training Materials

FPGA

1 Introduction

The First User Action (FUSE) is a technology transfer programme in microelectronics. It is aimed at stimulating the wider use of microelectronics technologies by European enterprises to improve their competitiveness and enhance their growth. FUSE is funded by the European Commission (EC) under the Information Technologies programme, ESPRIT, of the fourth framework. FUSE operates through a network of Technology Transfer Nodes (TTNs) covering all regions of Europe. The TTNs promote the first use of a wide range of microelectronics technologies and support enterprises in their regions as they acquire these technologies. FUSE utilises a portfolio of demonstrators to persuade enterprises to adopt new microelectronics technologies and to show these enterprises how to overcome the knowledge, technology, cultural and other barriers that have prevented them from utilising the appropriate technologies to achieve economic growth.

Field Programmable Gate Arrays (FPGAs) represent a class of digital devices within microelectronics technologies that enable a large amount of functionality to be integrated into a single device. Their functionality is capable of being programmed by the User and as such they offer potential Users the advantages of reduced component numbers, design security, increased performance and lower costs. Figure 1 shows some examples of PCBs with FPGAs.

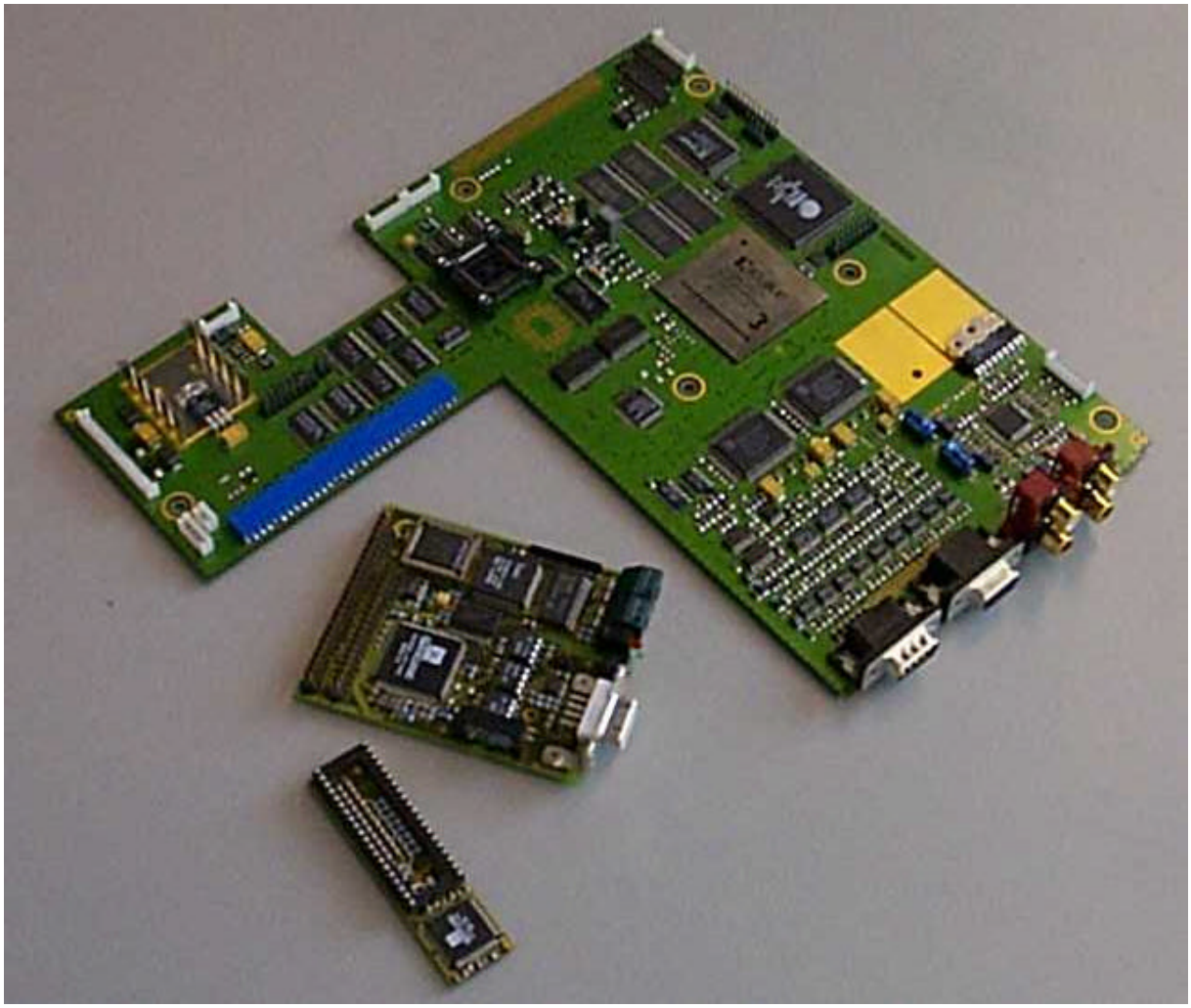

Figure 1 Examples of PCBs with FPGAs

1.1 What constitutes an FPGA?

There is no standardisation amongst manufacturers of programmable devices, and the terms EPLD (Erasable Programmable Logic Device), CPLD (Complex Programmable Logic Device) and FPGA are all used when referring to programmable devices. In this document we use the term FPGA to describe a device which consists

of an array of programmable blocks containing logic elements and flip-flops with a programmable interconnect matrix.

1.2 How is the programme stored?

Essentially there are two very distinct types of FPGA, and the difference is simply to do with way that they "store" the configuration information. The different types and the principal advantages and disadvantages are explained in the following table:-

Type	Advantage	Disadvantage
Memory based Devices These use Static Random Access Memory (SRAM) in order to programme either the logic block function or the interconnect matrix function. As a result of the use of SRAM, the function and interconnect properties will be lost once power is removed. It is therefore necessary to re-programme the device every time it is powered up	Can be reconfigured easily for different functions giving increased versatility. During development the same device can be re-programmed again and again. Can be programmed in circuit.	Requires means of loading the configuration data at power up. SRAM elements make device large.
"Fuse" based device These use some form of semiconductor fuse or antifuse element to store either the logic block function, or the interconnect matrix function. The device is therefore "one time programmable".	Can be "single chip" solution Non-volatile Device uses less silicon	Not in circuit programmable Requires programmer During development, a large number of devices may be wasted

1.2.1 Programming Facilities

As mentioned above, there are several FPGA technologies. The selection of this technology may be influenced by a number of factors and, in addition, different technologies are advantageous for different applications. This section discusses some aspects of FPGA technologies.

1.2.1.1 SRAM technology

If an FPGA is to be used for prototyping, SRAM technology provides the highest flexibility. The system may be modified step by step until functionality is satisfactory. However, this also carries the danger of bad design style since it results in an "ad hoc" approach to problem solving.

In numerical applications, different algorithms may be tested and modified with the minimum of delay. It may be an advantage to be able to run multiple applications on the same PCB, and this is possible provided the customer can change the FPGA configuration. The configuration of the FPGA may be done by the parallel or serial port of a PC, by a micro controller which is later used for further processing or by an EPROM which is automatically read by the FPGA at system start-up. The suitability of a common EPROM-Programmer for the specific FPGA and EPROM families needs to be checked. SRAM devices also give the ability to update the hardware in the field without having to remove the device from the board.

1.2.1.2 AntiFuse technology

Using Anti-Fuse technology is often preferable for high-speed and compact applications. This is because AntiFuse FPGAs usually are faster than otherwise comparable SRAM types and no configuration EPROM is required. They are only one time programmable, which may cause a higher investment during the development phase, but usually leads to a more thorough, ASIC-like design style. The programming of an Anti-Fuse based FPGA takes time, the actual amount of time depending on the programmer and the size of device being programmed. The programming must be done during the manufacturing process and requires investment in either the programming equipment or in the programming services provided by the manufacturer or it's distributor.

1.3 Creating the design

As for any project, specification of functional and non-functional requirements is the first step of an electronic design. It should describe all aspects relevant to the system under design; for example functionality, costs and the environment it is intended to be used in. The general constraints help to select appropriate target technologies and following this selection, the general constraints can be transformed into technology-specific constraints for the selection of specific design methods and devices. The next step is to divide functionality into manageable blocks and to define internal and external interfaces. It is good advice to select or define a set of design rules to guide the design process. This is especially important if a number of designers need to co-operate and helps to match design styles.

The design can be created using **Schematic Capture** or by using a **Hardware Description Language** (HDL) to define how the function is to be performed. During schematic capture, the required function is drawn using library elements supplied by the manufacturer of the FPGA to be used. This has the advantage of using methods, and sometimes tools, that are familiar to the User. Schematic capture directly models the circuitry to be implemented, but often undergoes automatic optimisation by design tools. HDLs allow the User to define the required function in text form, often starting with simply a description of the required behaviour. This initial "behavioural" description is then expanded to provide a detailed description of the required elements. It is then necessary to carry out a synthesis stage to convert this written description into actual hardware (i.e. a representation using gates and flip-flops which could be shown as a schematic), but it should be noted that the description must be written with this intention (that is synthesis) in mind or the process will fail. The majority of FPGA manufacturers provide guidance on this aspect, which is usually referred to as "coding style". Many of the design tools also check for design errors, such as unconnected components, design rule violations etc.

1.4 Simulating the design

It is essential that, regardless of which method is used, the design is simulated. That is a computer is used to model the actual operation of the circuit under pre-defined sets of conditions. Many FPGA designers, especially using SRAM-based devices, are tempted to neglect a thorough analysis of the design within individual design cycles. They find a solution, which is working more or less, only by chance. The resultant devices often cause problems with changing environment conditions, and these may eventually cause malfunctions. In order to assure the high quality of a design, simulation is a crucial prerequisite. For simulation a set of signals are created on the computer, and these signals are used as inputs to the circuit and the results are viewed to ensure that the circuit functions correctly. Since the actual interconnect which will be used is not yet known, this simulation cannot make allowances for the effects (in terms of delays etc.) that the actual layout will have on the performance. It is therefore essential to also carry out this simulation after the layout has actually been carried out (see Section 1.5).

1.5 Programming the device

Software is then used to convert the "schematic" into at least two pieces of information –

- the Layout (the required circuit in the form of the gates and flip-flops within the FPGA),

- the Programming information (which will either be sent to the SRAM based FPGA, or will be used to "blow" the fuses in Fuse based FPGA or connect wires with an antifuse)

As has already been mentioned at this stage it is also essential to re-simulate the design. This involves a process which is sometimes called back-annotation, where the effects of the layout (the interconnect as well as the actual blocks used for the logic functions) are taken into account and the anticipated delays can be used in the simulation. This will give a more valid picture of the operation of the circuit and enable an accurate assessment to be made as to whether the circuit meets all the performance criteria in the specification.

Which FPGA should I use?" and the final one is "Shall I use Schematic Capture or Hardware Description Language?".

2.1 Why use an FPGA?

Perhaps the best way to provide this information is to use a table showing the various technologies, and how they compare with FPGAs.

Technology	**Remarks**
Discrete Analogue* Components	Must be used where the function is and can only be analogue.
Discrete Digital* components	Fixed function – any change may need additional components and PCB changes Multiple components required, and therefore increased space, with more solder joints.
Mixed Signal ASIC	Applicable where there is a mixture if analogue and digital functions and normally where the annual quantity is in excess of at least 10,000 pieces. There is a high development cost – often called the Non Recurring Engineering cost or NRE. Development time can be long.
Digital ASIC	Normally applicable where the annual quantity is in excess of at least 10,000 pieces. There is a high development cost – often called the Non Recurring Engineering cost or NRE.
Microcontroller or Digital Signal Processor (DSP)	Applicable where flexibility is required – may be restricted by the ability to process the data in time. It also has to be observed that microcontroller and DSP only have a single or very few processing units, such that concurrent tasks have to be serialised for processing.
FPGA	Flexible – enables response to customer's requests to be made with minimum effort. Fast dedicated processing Some devices may be dynamically reconfigured. Suitable for low volumes – from 1 unit up to 10,000 Means fewer components. Provides a short "Time to Market" – typical figures derived from FUSE Application Experiments range from 5 months to 1 years.

* Signals in digital systems only know two values. Signal in analogue systems may have many more values in a range of a minimum and a maximum value, only restricted by the resolution of its devices, and these values may vary with time.

2.2 Which FPGA should I use?

The choice of FPGAs is very wide and the factors which have been taken into account when making the choice include the following :-

Criteria	**Comment**
Do I need to be able to change the function.	SRAM based FPGAs enable the function to be readily changed. Anti-Fuse based FPGAs however cannot have the function changed once they are programmed.
Do I require the smallest possible solution	Fuse based FPGAs may be able to provide a “one chip” solution
Do I need to be able to dynamically reconfigure my device.	Certain FPGAs (SRAM based) enable you to change all or part of the function “on the fly”
Will the device be obsolete soon?	It is essential to check on the availability of devices in order to ensure that you are not going to be forced to redesign your product.
What System clock speed do I need?	Ensure that the system clock will be suitable for your application. Check out the internal structure to make sure that you are aware of the actual “internal” clock frequency. Do rely solely on the quoted toggle rate for the flip flops. Also remember that layout can drastically effect the actual speed.
Is there enough I/O and is it suitable?	Ensure that the device, and the suitable package provide enough Some devices have registered I/O and this resource can be very useful.
Device Cost.	Check out the price of devices that are suitable.
Is the power consumption too high?	This is particularly important for battery powered applications – but can also be important if the thermal performance of the product is critical.
Is testability important?	Many devices include testability functions (e.g. JTAG) and this should be considered.
Is second sourcing available?	Compatible devices are sometimes available, and this should be considered.
Is the vendor reliable?	The FPGA market is extremely competitive, and it can be dangerous to go with a new vendor who may withdraw his product at short notice.
Are the features that I want to use well documented?	Do not take the word of the “salesman” but check out the documentation – particularly if the feature is important (e.g. dynamic reconfigurability).
Are the development tools suitable?	The tools should be fully integrated allowing the easy interchange of data between design input, synthesis, and layout. Can the netlist be exported for use in other software if required?

2.3 How should I create my design

That leaves the final choice – to use an HDL: or to use Schematic Capture.

Using schematic capture, elements of the circuit to be used are selected from a library and interconnected. Using HDL, the circuit is described textually with control operations like 'if, then, else' and data operations like '+, -, *, . The HDL description has to be transformed into a netlist, which defines actual components and their interconnections, before it can be loaded into an FPGA. This is usually done by a synthesis tool.

Schematic Capture	**Hardware Description Languages**
Not suitable for very large designs	Suitable for any size design
Enables re-use of existing schematic designs (note though that any design for an FPGA must follow the basic rules relating to synchronous design – and these may not have been adhered to in existing designs)	Allows a fresh approach to be taken
Restricted to the current target device since the design will consist of library parts for a particular device or family of devices	Future proof, that is the HDL is not technology specific since the actual target is not selected until the synthesis stage. Therefore the same HDL description can be used to target any manufacturers device, and also to target an ASIC process (note though that some small changes may need to be made to meet the specific requirements of the chosen target).

3 Construction of a Work Plan

This chapter shows how to organise an FPGA project. The FPGA projects supported in FUSE have shown that the structure of the projects is often the same. In general, projects consist of several tasks which are often subdivided into work packages. The most important tasks will be described and the arrangement of the tasks will be presented at the end of this chapter. The main tasks of an FPGA project are project management, training, specification, design, verification and documentation. Some of this tasks may, of course, be done in parallel, whilst others may require to be completed before any other task can be commenced.

3.1 Project Management

Project management consists of three different phases.

1. Project Initialisation - The project must be initiated, by selecting the project and designating the project manager.
2. Implementation - The next phase, that of implementation, deals with project planning and controlling. Key elements of this phase are budgeting, scheduling, risk assessment, monitoring and reviewing.
3. Termination - The last phase is the termination of the project. It describes plans for finishing the project, and would look at the success or otherwise of the project and at what happens with the results of the project.

The projects evaluated for this document only have the phase of implementation in common, because the initiation and termination are strongly dependent on the company 's objectives. As a part of project management the subsections project planning and controlling will be explained.

3.1.1 Project Planning

The project planning deals with the constraints of a project. Costs, time, available manpower and the risk are the main constraints. If a project is to be successful it is important that the extent of these constraints is known and that the plan takes them into account. Critical elements should be identified and the risks associated with the task assessed and understood.

3.1.1.1 Budgeting

The budget includes the personnel costs, costs for subcontractor services, for FPGA-training, for design tools as well as a computer system suitable for the design tools. Furthermore there may be several additional costs for the circuit-design, the PCB-layout and the manufacturing. There may also be a need for special test equipment, and this must also be considered. Figure 2 shows an example of a work place for FPGA development.

For most of the SMEs, development costs appear as a barrier, which has to be overcome by sound budgeting of the development project. In addition to personnel costs, the costs of reliable CAD-tools are often a problem for small companies as is shown in the following extracts of AE 264, 2173, 2193.

As a small company with only three employees, Applied Relay Testing must concentrate on the demands of day-to-day production, leaving little time for the longer-term investment in developing new technologies. Undertaking any new development incurs risk and this creates a barrier by making it unacceptable to allocate disproportionate funds against a possible negative outcome. Such financial barriers only serve to reinforce the knowledge and technology barriers further.(AE264)

The main barrier for us was the long time it would take to get through such a project like the digital video FPGA and flat panel monitor and the many mistakes we could make, when trying to get into this new area by self education (learning by doing). The financial barrier of course is connected to the time barrier. It is a risk to spend over eight man months of work into a new product based on new technology. (AE2173).

Small companies like VIEW POINT are extremely cautious to enter large development especially when using unknown tools. This is enhanced in our case as we have the knowledge of software developments which may be much more longer than planned.(AE2193)

Fortunately many of the FPGA vendors are now supplying the software needed to design with their FPGAs at zero cost – however it must still be remembered that there will be a cost in terms of personnel as the staff become familiar with the particular software tools. It should also be remembered that the software provided by the FPGA vendors will often "lock you in" to their particular FPGAs.

The FPGA-Projects evaluated for this document had a budget of between 30 KECU and 70 KECU with the average budget being 50 KECU. AE 2197 and AE 23175 show a distribution over the tasks.

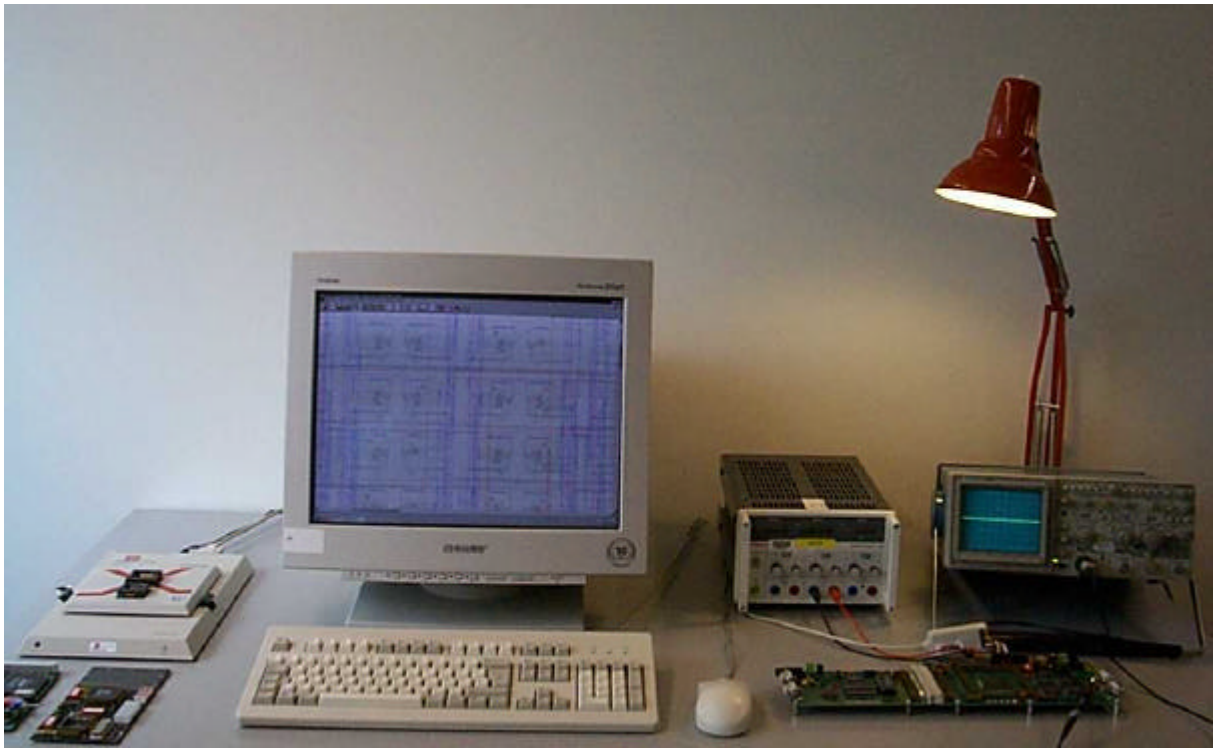

Figure 2 Example of FPGA design work place

3.1.1.2 Scheduling

Scheduling is the arrangement of the tasks according to time. Some of the tasks should appear in a special sequence. The FPGA design is often a only part of a whole system, and in order to do the system design a specification of the FPGA is required. However since the technological properties may not be known for the first FPGA based design, the FPGA may be designed first. Afterwards when starting the remaining system design, all the information on functionality, pinning and timing as specified for standard chips is available. FPGA-Design should therefore be considered as a separate task with a milestone at the end.

Planning a schedule should also take care of deviations for day-to-day business and 'sudden' customer requests that may require to pause a development project. However, the timing of the development project, e.g. date for introduction into the market, has to be observed.

Problems in scheduling and the long time of development were mentioned as a barrier in the AE 2193:

Arrange the planning so as to be able to face normal work in the company and development needs.

For scheduling purposes the runtime of synthesis tools (required if HDL is used) should be taken into consideration. This is especially true in the phase of verification where the whole design has to be synthesised after any changes. Each a synthesis run may take several hours, especially if the design is not a very simple one. Since designing is an iterative process a high number of such synthesis runs may be required. An expert may give useful hints on the overall time required for a specific project.

The projects carried out in FUSE showed an average project for a PCB-based design with FPGA technology will take about 9 months.

3.1.1.3 Resource Allocation

The scheduling of a work plan is closely related to resource allocation. It has to be done with great care, especially if the personnel are also employed in other projects and the equipment is used for other tasks. The resources should include the person-hours of special engineers as well as of technical services, computing time and machine-hours for manufacturing. Consumables and specific software may also be required.

FPGA-Projects specifically require engineers who are familiar with digital circuit designs for creating the FPGA-designs as well as for the system designs. Designing is not merely an add-on to daily business but a full-time job. For an effective design review to be carried out, including the verification of design results, at least one other person ought to be involved in the project. Further personnel may also be required for project management, documentation etc. The PCB-layout and the manufacturing may either be subcontracted or can be done in-house and in the later case appropriate resources need to be allocated, with an allowance made for design iterations.

Quality of design platform and software should be observed. A 21-inch-monitor is recommended for schematic entry. In many projects, the poor performance of design platform and tools resulted in a need for changes. It is therefore worth evaluating the tools first or talking to an expert who has used a variety of different CAD tools.

3.1.2 Risk Assessment

During the phase of planning, the risks associated with the project should be considered. There are different classes of risks, especially

1. technology and tools
2. customer
3. subcontractor
4. project team
5. project management

3.1.2.1.1 Technology and Tools

The new technology is obviously the item with the highest risk for a First User (FU) of an FPGA project. The FPGA-devices as well as the software-tools are unknown to them. An investment is necessary to get reliable tools and the return on this investment is often uncertain during the planning phase, because the FU cannot be sure of the suitability of the tools.

The selection of the FPGA vendor is one problem, the selection of the CAD-tools is another one. THARSYS (AE182) describes such decisions as a risk for a FU. They don't know:

1. *in which type of microelectronics they should invest because the devices of Lattice, Xilinx, Altera are apparently similar,*
2. *what is VHDL, and which benefits do they expect from this methodology,*
3. *which design methodology should be used (schematics, logic equations),*
4. *which development tool is the most appropriate - each one being more friendly, reliable and efficient than the other, according to each software house marketing - and which costly options have to be purchased.*

THARSYS chose a subcontractor as a consultant in order to answer such difficult questions. Consequently no problems with the FPGA-selection and programming tools during the project were mentioned.

The Application Experiments have shown that CAD-tools were often changed or combined with tools from other vendors during the project. If this happens then a significant number of alterations may be required in order to transform the design to the new software tool. Personnel cost may well increase initially, even though overall costs may eventually be lower. Incompatibilities among different tools as they are described by BEDE TECHNOLOGY Ltd. (AE 2197) are typical for using a mixture of tools:

The Actel Designer Series software was extremely good as a stand alone product but we did have problems integrating it with the other CAD tools. The major problem was with bi-directional data busses. Our original version had a software bug which caused busses which were synthesised by Synplify to be reversed. We worked around this for most simulations by using extra VHDL code to reverse these busses for display by V-System. As we were using three different suppliers for our VHDL CAD tools the cause of this problem was initially very difficult to track down. However, we eventually received a new version of Designer which fixed this bug.

3.1.2.1.2 Customer

For a company in the service sector, it is crucial to know what the customer wants. Changes during the project are expensive and will be the basis for conflict. A specification, agreed on and signed by both parties is essential in order to minimise any conflicts. If a customer is not able to completely specify his requirements and the system to be designed, the task of specification should be offered as an additional service. Later changes of specifications have to be done with great care. Customers tend to play down the importance of changes, yet any change can be critical for costs and schedule of the project. Revision of specification during the project also often requires revision of the schedule and the costs.

For a company in the product market, it is equally important to have a detailed concept of the product to be designed, its implementation and production during its entire lifecycle. Without a specification, the design task will turn into a trial and error situation with uncertain outcome. This may cause a redesign of PCB-layouts, and effort and costs may increase dramatically. This typical problem is often one effect of insufficient project control. A strictly executed out review will spare you this situation.

FN Herstal (AE 24683) said they had difficulties in specifying their needs at the beginning of the project, but with the help of a subcontractor they organised their work according to plan.

...our people did not have the technical knowledge required to specify, design and test circuits using the FPGA technology. (AE177)

3.1.2.1.3 Subcontractor

Many companies include a risk by selecting the wrong subcontractor. They do not know how to evaluate its abilities and to control the quality of its services. To reduce the risk many First Users have chosen an Institute or a University as a subcontractor for design support as well as a consultant in technology and design tools. Important aspects of selection are independence of the subcontractor and expertise on a certain application area.

The costs for the training are considered as a risk by the company Wittenstein Motion Control GmbH (AE2239) Often the training is not a measurable service and it is difficult to evaluate. The selection of subcontractors is handled in more detail in a subsequent section.

3.1.2.1.4 Project team

The company View Point (AE2193) mentioned the motivation of the project team as a risk. Personnel are required to do something new and may feel fully responsible for the outcome and yet not have full control. This is especially true if the project team does not accept the new technology or the design tools (for example because of doubts about reliability of the new technology) problems of motivation may occur as it is shown by FN Herstal (AE24683).

3.1.2.1.5 Project management in general

Project management may not be a standard procedure for all FU, because they have no experience with structuring their work in such a manner or consider it as a formidable action.

Applied Relay Testing (AE264), a company with 3 employees said after the project:

> *With the benefit of hindsight, the task of producing a suitable Workplan with the required milestones and deliverables was a formidable one. Applied Relay Testing was one of the first applicants to be approved for a FUSE experiment and this resulted in a high level of forward*

planning having to be applied at an early stage. The extent of this forward planning is unusual for a small company and the amount of work involved was unexpected, however due to the support and encouragement of our TTN (Bournemouth University) the Workplan was adjusted to include realistic and manageable deliverables. The benefit of this early work was that during the implementation of the Workplan, the attention to detail and accuracy resulted in a very controlled and measurable project which was easy to monitor and report. Due to the amount of early planning and thought which went in to the Workplan, where small changes had to be subsequently made to time-scales, these were then easily accommodated by simple discussion with our TTN and the relevant adjustment to the plan without actually changing the nature of the deliverables.

3.1.3 Project Control

After establishing a project structure, plans should be repeatedly evaluated against the reality. For this purpose, milestones should be marked in the work plan. A delay in any one work package or rising costs should motivate the project manager to act, and take whatever action is appropriate.

The Application Experiments have shown that there was often insufficient project control. Reviews didn't take place or the conclusions weren't drawn, and many First Users didn't spend time on project management. Writing a good specification or a documentation was considered as unnecessary work by many First Users. However at the design review there should be no leniency with regard to missing documents.

Information for your work plan:	
Work packages	project planning, project control
Milestones	project start, review dates and project termination

3.2 Training

The training in a FPGA project can take place in various ways. For companies that are not familiar with project management of FPGA projects, a subcontractor should be asked to act as a consultant. Writing a specification should be done with the help of a subcontractor and a subcontractor may also provide basic information on FPGA technology and training design. An introduction to the design tools should also be received from an appropriate source. However, good training should also consider the existing skills of the first User, and for this reason a training provider who can “tailor” the training to meet the particular needs of the First User may be preferred.

During training, the First User should develop an FPGA design flow, suitable for its situation. Training on the project provides a way to gain fast access to the new technology and reach the objectives. This is especially important if hardware description languages (HDL) are used. It was very helpful when making the selection of tools to get a demonstration of different design tools. The section *Management of Subcontractors* gives further hints.

The training material handed out by the subcontractor as well as the notes taken by the trainee should be filed within the company and will be important if the trained employees leaves the company. It is good practice to provide training to more than one person, which may be done by internal training or on the project training such that external costs are minimised.

The training may be split into several sessions in order to minimise the disruption to the company’s operation.

3.3 Specification

This is divided into a general specification and a technical specification. The general specification includes a description of the objectives of the project, the constraints under which the project is running. The technical specification is about the functionality of the system. This technical specification will then be divided into a number of sub-system specifications for each distinct module on the system, of which the FPGA will be one. This is shown in the following example:

> ***Example**: Considering a PC slot-card with master functionality on a local area network system. The master functionality will be your whole functionality. As components, you'll have the interface to the net, the time-critical protocol stack will be realised by programmable logic devices and the upper layers should be realised with a micro controller. A dual ported ram may be the interface to the PC-slot. After you have defined the interface of each component, the part that is selected for the programmable logic device can be started in a separate specification. The functionality should be recursively divided until the lowest level is specified. Classical design techniques such as the distinction between control and data flow gives you further hints on a good design style. At this point you have the basis for starting your design. This kind of design style is called a top-down.*

A specification is more of a functional description then an architectural-design. Many people consider coding as a specification, but it is in fact realisation. The specification describes what tasks have to be done, how these tasks were realised will later be fixed in the documentation.

In addition to the written specification, there may also be block-diagrams, flow-charts, Nassi Shneiderman diagrams or automata tables etc. as well as a test specification which defines the test requirements.

The specification for the FPGA should contain the following information:

Information for your work plan	
Work packages	multiple sessions of training on different subjects
Points for reviewing	document about what was done during the training phase
Milestone	Each training session should be terminated with a milestone.

1. technology of the FPGA
2. family of FPGA
3. list of all signals, the actual pins may not yet be allocated
4. functional block-diagram
5. hierarchy of the modules
6. interfaces of the modules
7. coding styles
8. check-list for good design style and rules

The technical specification for the FPGA should also include the description of a testbench, the code which must be written for testing the design.

The technical specification of the PCB should describe the set up of the PCB.

Information for your work plan	
Work packages	one for the whole specification
Points for reviewing	document in bookstyle
Milestone	completeness of the specification.

3.4 Design

The design is a computer readable document, which is the result of the transformed specification. The transformation can take place in several ways:

1. Generating a program in a hardware description language (HDL-entry)
2. Drawing the circuit with standard elements (schematic entry)
3. Using building blocks from other vendors as Intellectual Property
4. Using code generators especially for state machines
5. A combination of two or more of the above

After the specification has been done in a top-down-technique, the design must be realised by bottom up implementation. The migration from standard devices to FPGA-technology is often more than merely changing the technology, because the FPGA demands a synchronous design. Therefore changes to the original circuit-concept may be necessary. A subcontractor should provide training on synchronous digital design, especially the basic elements of circuit design should be transformed in their HDL-equivalence. When using an FPGA synchronisation of several clock domains may be necessary. Methods for solving this problem do already exist and should also be explained by the subcontractor.

The phase of coding or schematic entry should be guided by the *Golden Design Rule* (see Appendix A). They describe what a design engineer should do or should not do for a reliable design. For coding it could be very useful to set up a document where the coding style is fixed. Verification, especially simulation, is carried out during the coding because the sub-designs should be verified step by step.

For setting up a testbench a description of the periphery of the FPGA is required. There are different ways for generating a testbench. A lot of tools provide a graphical interface where waveforms, as a stimuli for simulation, may be drawn. The outputs of the simulation will also be displayed as waves. If stimuli consist of several sequences the description in an HDL could be more comfortable, e.g. using loop-functions. Check the simulation tool to see if a proper HDL entry is supported.

During the implementation there are often new ideas, but be aware, not all additional features have to be realised in the first design, the ideas should be noted for further releases. After successful simulation of the top level the complexity of the design can be determined. The device of the FPGA family and a package can also be selected. A pin list can either be generated automatically, or manually by assigning signal to pins. Assigning pins manually is a constraint which can adversely affect the layout in the same way that manually placing components when laying out a PCB can affect the layout software. A very important tool is a timing analyser for extraction of timing information. After this procedure all information, required for system integration is available.

1. A complex FPGA design may be subdivided into several work packages, because the plan is easier to handle.

2. The overall system, of which the FPGA is a part, has to be designed. For prototyping it is advisable to use sockets for FPGAs. This is because you will almost certainly have to programme and fit different devices as you test your design.

3. If a microcontroller is used in the system, software design is an additional work package.

Information for your work plan	
Work packages	FPGA design, peripheral circuit design, software, if necessary
Points for reviewing	list of coding-styles, listing of source-code, simulation-results, timing diagram, pinning and device selection according to package and speed grade, circuitry.
Milestones	The programming-files for hardware as well as for software of the specified designs according to the reviewed items and the circuitry have completely be finished.

All packages should finish with a milestone.

3.5 Prototyping

3.5.1 Programming the devices

Antifuse-FPGAs require a programmer, similar to an EPROM programmer for programming of devices. The programming time is in the range of several minutes up to an hour. This time may not matter in the phase of prototyping, but may be critical during production. Some FPGA vendors or their distributors offer special programming services and with this special service you can save the investment on the expensive programming tools. SRAM-FPGAs, may load their configuration from an EEPROM or they may be programmed by a micro controller. The code for the FPGA and the code for the micro controller can be combined in one EPROM. Alternatively, for test purposes, they may be programmed via a serial or parallel link from the PC that is being used to develop the design on, although this will obviously not normally be the case for production units.

3.5.2 PCB-Layout

For FPGA technology a multilayer PCB is often necessary due to the density of the pins. Most of the packages are SMT devices and this may be the first time that many companies have used SMT devices. . Some are even available in fine pitch technology (< 0.6mm from pin to pin) and as Ball Grid Arrays, and this may be an advantage where space is at a premium. For fine pitch technology the surface of the PCB must be plane and free of solder, so that you are able to put the solder paste exactly on the solder points. This kind of PCB is more expensive and they are not offered by all manufacturers. You should order PCBs that have been electrically tested by the manufacturer in order to remove one possible source of faults. Furthermore, the restrictions of automatic production must be considered. With respect to using sockets for prototyping, two PCBs, one for prototyping with an FPGA socket and one for production (without the socket) may be required.

3.5.3 Manufacturing

In general, First Users should be familiar with manufacturing PCB. Manufacturing, using static sensitive packages should be done automatically. Placing devices manually may be an exception if you are using PLCC-packages.

Information for your work plan	
Work package	Device programming, PCB layout, manufacturing
Milestone	PCB layout, receiving the PCB from the manufacturer, assembled PCB

3.6 Verification

The testing of the PCB must be done on the basis of the test specification. The results of the tests should be fixed in a protocol. Individual components of a system, like hardware and software should be tested separately prior to test of the complete system. The final verification must take place in the actual environment.

Information for your work plan	
Work package:	hardware test
Review:	test protocol
Milestones:	the measured values match the specified ones.

Testability: there should be no structures that cannot be verified.

3.7 Documentation

The documentation describes how the design was done. The FPGA is described with details of the sub-modules - a textual description as well as the architectural design is presented. The documents of the FPGA review such as

pinning and timing diagram must be included. The circuitry, the layout and the data of the manufacturing must be added. The document is finalised with the test protocol.

3.8 Arrangement of the work packages

Gantt- respective bar-charts provide good visualisation for a work plan. All work packages are shown as bars with their planned duration and the offset from the project start. The review meetings and the milestones should also be included. The actual dates of the workpackages should be added as separate bars. This chart therefore gives an overview of the project's position at any time.

A theoretically constructed Gantt-Chart is shown in Figure 3. At the beginning of the project a phase of project planning takes place. At the same time a subcontractor starts the training of the management skills. Usually, the training session is shorter than the FU's planning phase. The construction of the specification is arranged in the same manner: At the beginning of the specification phase there is also a support by a subcontractor. The design phase is split in three work packages: the FPGA part, a system respective circuit part and a software part. There is only a small overlapping in the work packages because one is based on the other. After finishing the FPGA-design the programming of the FPGA may start. As it is shown, the delay of the design phase propagates in the phase of prototyping. The PCB layout starts with the end of the circuit-design and there is a gap between PCB-layout and manufacturing which may be required for procurement of components. The evaluation of the components starts with the commissioning of the hardware although some features of the hardware may only tested by software as indicated by the overlap of the work packages. From the beginning of design phase until the end of the evaluation the phase of documentation will take place. Reviews are organised monthly during the whole project.

> **Example:** The company Meilhaus Electronic (AE177) has more than 20 years experiences in data acquisition systems. Their portfolio includes different interface cards and conversion devices.

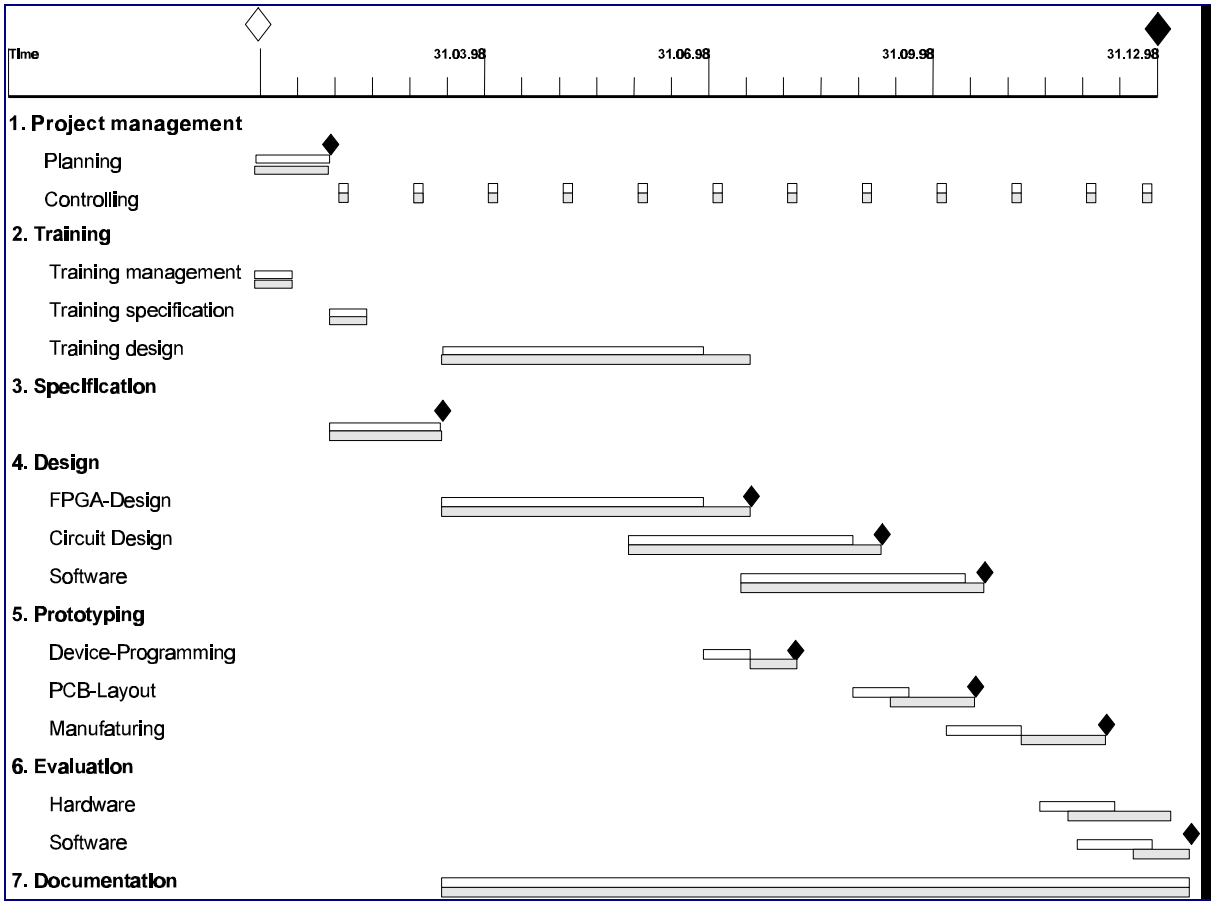

Figure 3: Gantt-Chart

Development software is also available. So the skills of the company's personnel were in hardware as well as in software. For generating their PCB-layouts they were familiar with CAD-tools for schematic entry. They used PAL and GAL components before they started the AE. The company has had no experiences in designing FPGA or CPLD

	Effort FU	Effort Subcontractor	Total
Project Management and Reporting	40days	30 days	70 days
Specifications	35 days	--	35 days
Training	20 days	20 days	40 days
Design	220 days	--	220 days
Evaluation and Testing	75 days	20 days	95 days
Total	390 days	70 days	460 days

The AE had a planned duration of 12 month. The actual duration was 16 month due to some technical problems. The integration of a PCI core in an FPGA couldn't be realised due to the lack of stability of the core. They eliminated this problem with standard components. This change, in addition to several faults with the CAD-tool caused the delay of 4 month.

Example: The company Applied Relay Testing Ltd. (AE 264) is in the market for relay test equipment. Since 1992 they have developed test systems and the three employees designed their products with hardware as well as with software components. Only standard components have been used in the past, and they had previously carried out only very basic project management. To improve their test systems they needed more information on digital signal processing with DSP and FPGA / CPLD programming. The work was done within 9 month, one month less than originally planned.

Task 1 Project management

During the whole phase of the project the company's manager was involved in planning and implementing as well as in reporting and monitoring.

Task 2 Specification

The specification consists of the development of multiple DSP algorithm and the selection of the most suitable CPLD and MOS-(Metal Oxide Semiconductor)-devices

Task 3 Training and preliminary engineering investigation

The training was subdivided into 3 parts. For the digital signal processing they used subcontractor information to brush up their knowledge. The behaviour of the MOS-devices was examined in test-sequences by themselves. For programming PLD, they used also a subcontractor who trained their skills.

Task 4 Design work

The design work includes the following parts. The PCB with MOS-switches was designed.. A logic PCB was designed based on the CPLD and a PCB with DSP was designed for implementing the algorithm. All parts were connected to the final system. Whilst the designs were done in-house, the PCB manufacturing and assembling was subcontracted.

Task 5 Evaluation

During this task each component was tested on its own and in connection with another. The measurement data was documented in a report.

	Effort of the FU and the subcontractor
Project Management and Reporting	15 days
Specifications	21 days
	24 days
Design	
Evaluation and Testing	27 days
	132 days

Management of a Subcontractor

Why should I use the Support of a Subcontractor?

great assistance. they will assist the First User in distinguishing between the important occurring in the world of FPGA technology.
knowledge on the new technology and gains self confidence. risk of the project will be reduced furthermore, the . The failure in a project in which technology is applied for the first time will cause an even higher psychological barrier for future FPGA projects

4.2

A company that is familiar with project management in general will only need some information on what is typical for FPGA projects and where may be .
the first time may also advisable. should already have experience with consulting companies of the First Users size.

for example, many do not
. First Users who are not familiar with specifications often consider it as an in itself For some people it is more a psychological problem, because they become aware of the lack of information they have about their circuit For people who are advanced in project management, it is very useful these missing items such an early point in then correct shortcomings However, people are often afraid of taking a ist areas show a lack of control Some assistance in the area of specification may give an FPGA design a good start.

. In order to
the design
received. will enable more universal expertise acquired. designs created
competent partners is often very helpful The special issues
integration of to the remaining circuitry should be explained by the subcontractor Subcontractors with much experience in the area of FPGA-technology will be able to demonstrate different types of synthesis tools . There may also be advice on selection of a design tool for the

4.3

On the market there is a number of companies that offer training on programmable logic devices as well as contract FPGA designs The selection of the right one is often difficult.
with experience in designing programmable logic devices and possibly in the application domain of the First User The number of FPGA projects done on different technologies and on different tools

and reliability of his work. The subcontractor who provides training on project management and on specification should also be familiar with FPGA technology. This is because people who have already worked with FPGAs can explain the different work packages relating to the structure of the specification. It is not necessary that the manufacturer of the PCB knows FPGA technology.

4.4 Management of the subcontracted work?

The co-operation with a subcontractor, based on a commercial agreement, should include all the work packages where the subcontractor was asked to provide help. It should be confirmed that the work packages are in line with the phases of the project. It is recommended to start the training after basic specification, containing only application-specific aspects, has been finished. This way, the First User can provide the subcontractor with a good list of requirements and may compare quotations much more easily. Contracts must clearly state responsibilities and the measures of success. First User should try to find measures within their own application domain, such that they can verify the subcontractors results.

Training should be arranged in several training sessions. During training on the project, each session may be split into a presentation phase that is done by the subcontractor, the transformation on the application that is done by the First User and the discussion. Each session should be measurable and never be confused with the learning that takes place at a school or college.

Golden Rules

.

Hierarchical design	the pad level consists only of the pads and the top level	
Synchronous design	no combinational feed back loops no pulse generation using propagation delays synchronising asynchronous signals no timing violations of setup and hold times	
fan out	no overrun due to the fan-out limits	
signal names	use only alphanumerical characters and underlines members of a group should be numbered use the same name in all hierarchies use at most 12 characters for each name use self-explaining names	
clock signals	clock all flip-flops with the same signal don’t use combinational logic in the clock path don’t connect a data-input of a Flip-Flop to the clock don’t generate your clock with combinational logic	
global reset	connect all Flip-Flops to an asynchronous reset, for initialisation be aware of controllability be aware of visibility	
further checks	Avoid floating inputs avoid deep sequential structures	
documentation	Functional description of each module sign your changes with name and date	

Appendix B. Glossary of Terms

This is not intended to be an exhaustive list, but attempts to pick out certain key words and phrases which a may be unfamiliar to newcomers to FPGAs.

Anti FUSE	A semiconductor element which can be "blown" to create low resistance permanent link. Used in some FPGAs to store the function of a logic block or of the interconnection matrix.
ASIC	Application Specific Integrated Circuit. A integrated circuit whose function is designed for a particular application and a specific customer. Digital ASICs (or the digital portion of a mixed signal ASIC) require a strict and rigorous design style.
CPLD	Complex Programmable Logic Device
Dynamic reconfigurability	The ability of a programmable circuit to have its function re-defined during normal operation.
EPLD	Erasable Programmable Logic Device
EPROM	Electrically Programmable Read Only Memory. A re-programmable memory device used for storage of programs.
FPGA	Field Programmable Gate Array. An array of programmable blocks (containing gates and flip flops) with a programmable interconnect matrix
HDL	Hardware Description Language – a means of defining the function of a system in textual terms
JTAG	Joint Test Action Group. Group of organisations that defined a built in test strategy for use in digital electronic systems
Library Elements	Circuit elements defined by a manufacturer or CAD system which are designed to perform basic and complex functions (gates, flip flops, counters, adders, etc.)
Netlist	Description of a circuit (normal in text form) using it's basic elements (such as gates and flip flops) and including a description of the interconnections
Schematic Capture	The creation of a circuit diagram on a Computer Aided Design (CAD) system using standard symbols for circuit elements
SRAM	Static Random Access Memory. Semiconductor element that can be used to store information. Used in some FPGAs to store the function of a logic block or of the interconnection matrix.
Subcontractor	An individual or organisation which undertakes to work on behalf of or in assistance another.
Synthesis	The conversion of a textual based description of a system into a network of gates and flip flops that may be implemented in hardware.
Testability	The ability of a circuit to be tested either for faults of for function.