

Laboratório sobre a Implementação de Sistemas Digitais com HDLs Captura, Validação, Síntese, Implementação e Teste

Prática: Ativação dos leds e displays da plataforma XS40/XST-1

Recursos: Ambiente de Desenvolvimento Active-HDL da Aldec, Inc, CAD Foundation e Ferramentas XSTools da Xess, Inc.

Parte I – Introdução e Objetivos

O primeiro Laboratório sobre HDLs serviu para introduzir conceitos básicos sobre a utilização do ambiente de simulação Active-HDL, da empresa Aldec, bem como para realizar a simulação de um circuito exemplo, um acumulador de somas.

O objetivo específico do presente Laboratório será o de investigar um fluxo de projeto completo a partir de uma descrição inicial em linguagem de descrição de hardware, desde a captura do projeto até a sua implementação em hardware. Isto será feito mediante um exemplo simples, que deverá ser capturado, validado funcionalmente via simulação e sintetizado para a plataforma educacional XS40/XST-1 da empresa Xess. O exemplo deverá então ser descarregado na plataforma e testado. Finalmente, um exercício de alteração do projeto original, sua validação, síntese, descarga e novo teste devem ser realizados pelos alunos. Veremos que o trabalho neste Laboratório somente será possível pela utilização concomitante de dois ambientes vistos até agora na disciplina, o sistema de projeto auxiliado por computador (em inglês, *Computer Aided Design*, ou *CAD*) Foundation, da empresa Xilinx, e o Active-HDL, da empresa Aldec. Adicionalmente, este Laboratório apresenta uma breve introdução a FPGAs, os dispositivos sobre os quais as plataformas de prototipação usadas em aula estão baseadas.

Parte II – A Integração dos Ambientes Foundation e Active-HDL

Embora os ambientes Foundation e Active-HDL sejam produtos comerciais totalmente independentes, um acordo entre a Xilinx e a Aldec faz com que eles possam funcionar de forma integrada. De fato, o Active-HDL pode ser acionado diretamente a partir do gerenciador de projeto Foundation (via interface gráfica). Uma primeira questão que pode surgir é o motivo pela qual a integração entre o Foundation e o Active-HDL seria desejável. A resposta reside no fato do Active-HDL ser um ambiente de simulação apenas, sem capacidade de sintetizar hardware. Por outro lado, as ferramentas do Foundation permitem capturar, validar e sintetizar hardware a partir de esquemáticos, máquinas de estados finitas e geradores automáticos de módulos (LogiBLOX). Quanto a HDLs, o Foundation inclui ferramentas apenas para capturar e sintetizar hardware a partir de uma das linguagens de descrição de hardware: Verilog, VHDL ou Abel. Ou seja, o ambiente Foundation, por si só não possui capacidade de validação funcional de projetos capturados mediante uso de HDLs.

Uma segunda questão seria a motivação para separar ambientes de validação funcional (simulação) de projetos em HDLs de outros conjuntos de ferramentas. A vantagem disto é que a empresa que produz o software de validação (Aldec, no caso) não vincula sua ferramenta a nenhuma família de hardware específica, sendo geral. Isto aumenta as chances de seu produto atender um mercado maior. Note-se que acordos entre Aldec e fornecedores de hardware tal como Xilinx, Altera, IBM ou Intel, permitem que o usuário que deseja sintetizar hardware a partir de VHDL ou Verilog (apesar de não mencionado, o Active-HDL, a partir da versão 3.5 é capaz de simular VHDL e Verilog) possa dispor de uma única ferramenta de validação de seus arquivos VHDL. Além do mais, a complexidade técnica de realizar simulação em VHDL é extremamente alta, justificando o alto custo de um software como este, limitado a realizar apenas captura e validação de projetos em HDLs.

A partir do Foundation, é muito simples chamar o ambiente Active-HDL. Basta usar a opção de menu Tools → Simulation/Verification → Active-HDL Behavioral Simulation, que apenas está habilitada caso o Active-HDL esteja devidamente instalado e configurado para integração com o Foundation (o que é o caso no ambiente do Laboratório de Arquitetura de Computadores e Sistemas Digitais). Também é possível configurar o gerenciador de projeto para chamar o simulador Active-HDL quando se usa o botão de chamada do simulador (por omissão, este botão chama o simulador de portas lógicas). A reconfiguração do Foundation pode ser feita via a janela de diálogo de configuração, alcançada via opção de menu File → Preferences → Configuration... Na janela que surge, basta clicar na caixa ao lado da opção Use Active-HDL Simulation, caso esta já não esteja marcada, fechando esta janela a seguir com o botão e confirmando a escolha, caso apareça uma janela de diálogo para tanto.

PARTE III - Introdução a Dispositivos FPGA

Esta parte apresenta uns poucos conceitos básicos sobre FPGAs e sobre como descrever a relação entre sinais de entrada e saída no seu projeto e pinos reais do FPGA. Esta última parte é apenas uma revisão de assunto já visto na parte de projeto com esquemáticos.

1. FPGA – A sigla vem do inglês, e significa Field Programmable Gate Array

- **Definição** – Trata-se de um dispositivo eletrônico (“chip”) configurável pelo usuário (donde a palavra Field no nome, que significa No Campo, significando no campo de aplicação), através da descarga (*download*) da sua funcionalidade por algum meio (gravador de EPROM, cabo serial ou paralelo, leitura de ROM/PROM/EPROM na própria placa do dispositivo, etc.).
- Utilizamos FPGAs baseados em RAM da família Xilinx XC4000XL, cujos membros são compostos por várias partes configuráveis e reconfiguráveis, entre estes os blocos lógicos, as conexões entre estes e a interface com o mundo externo (ver Figura 1 abaixo).

FPGA

- **CLBs: Configurable Logic Blocks** – os blocos configuráveis.
- Interconexão configurável: permite conectar blocos configuráveis entre si.
- IOBs: interface configurável com o mundo externo. Cada pino pode ser configurado como entrada, saída ou ambos.
- **Visão Interna de um CLB:** bloco lógico configurável, contendo 2 tabelas-verdade configuráveis (*Look-Up-Tables* ou *LUTs*) como qualquer função Booleana de 4 variáveis (LUTs F e G) e uma como qualquer função de 3 variáveis (LUT H), podendo as 3 juntas serem configuradas para implementar uma única função de 5 variáveis. O CLB contém também 2 flip-flops e lógica de propagação rápida de vai-um para circuitos aritméticos.
- Exemplo: o dispositivo XC4005XL da placa XS40 possui 196 CLBs. Segundo a Xilinx, este *chip* pode implementar circuitos com o equivalente a algo entre 3000 e 9000 portas lógicas. (Exemplo: o processador Cleópatra gasta 170 CLBs do XC4005XL)

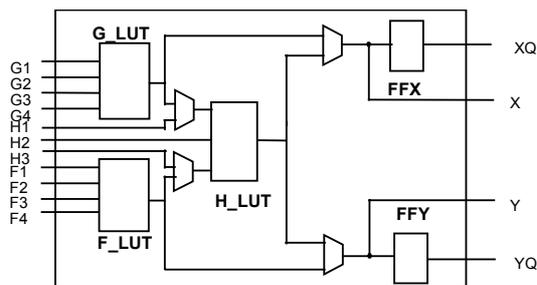
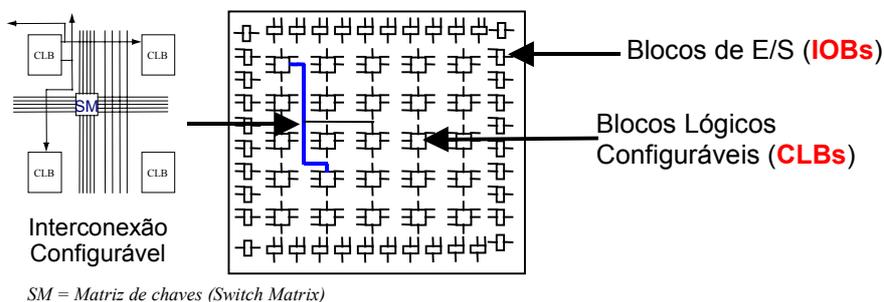


Figura 1 – Arquitetura de um Dispositivo Reconfigurável FPGA da família Xilinx XC4000XL.

2. Pinagem

- É responsabilidade do usuário definir a interface de seu circuito com o mundo externo, através da atribuição dos sinais de sua entidade mais externa (*entity*) aos pinos do FPGA.
- Consultar o esquemático da placa XS40 nos manuais da plataforma XS40/XST-1 para informação sobre as posições dos pinos específicos e suas conexões a periféricos das placas.
- O usuário entra com a informação do número dos pinos no arquivo com extensão *ucf*, na raiz do *Foundation*.(ver Figura 2). Sempre adicionar este arquivo ao projeto em desenvolvimento.
- Exemplo da parte final do arquivo *ucf*, onde são descritos os pinos de E/S:


```

...NET LED<3> LOC = P38 ;
NET LED<4> LOC = P35 ;
NET LED<5> LOC = P81 ;
NET LED<6> LOC = P80 ; .....
            
```
- Observar a relação destes pinos com o esquemático do manual. Estas linhas do arquivo de pinagem descrevem a conexão entre o barramento LED do exemplo (que ver-se-á a seguir) com os pinos dos leds da placa (no esquemático denominado de *Bar LED*, ou barra de leds).

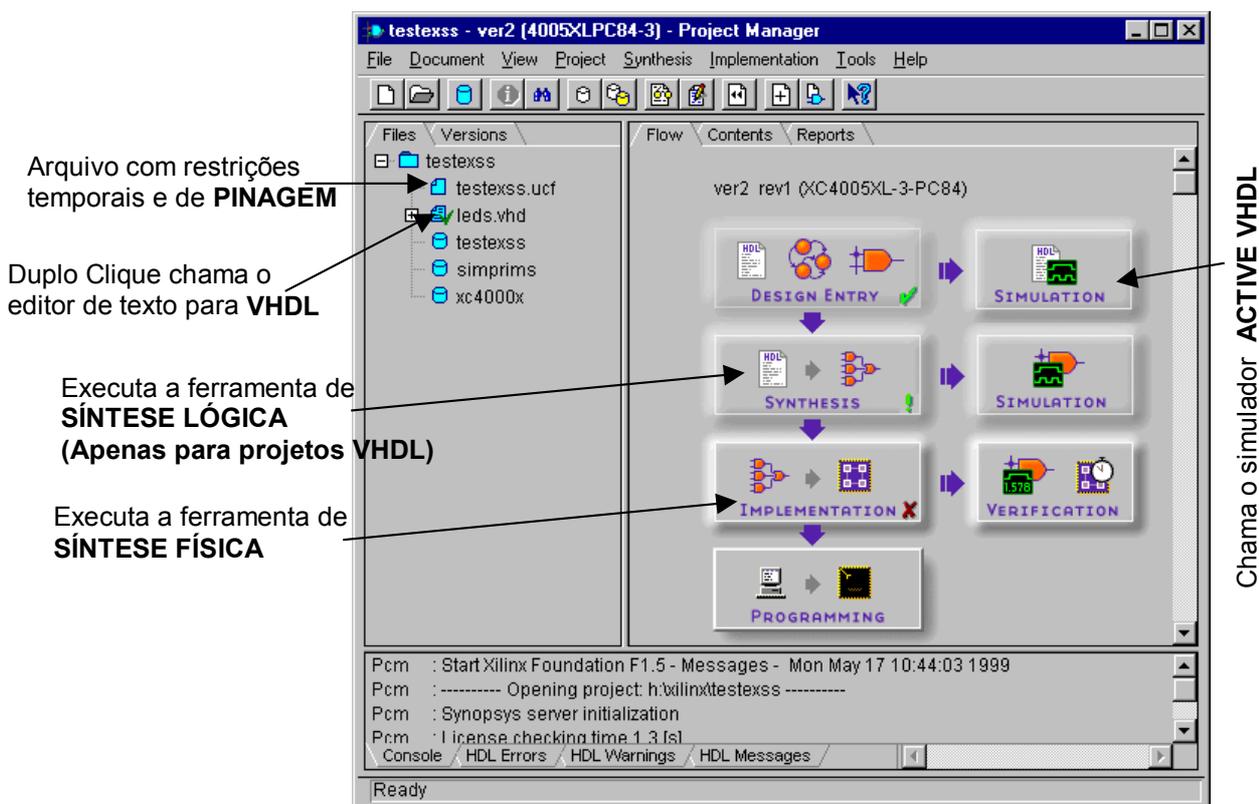


Figura 2 - Janela Principal do Foundation, com configurações de projeto VHDL salientadas.

Parte IV - Conceitos Fundamentais de Ferramentas de CAD

Nesta Parte, introduz-se alguma terminologia associada com o projeto de sistemas digitais, em particular o projeto via uso de HDLs.

3. Síntese Lógica

- Processo de transformar a descrição VHDL em uma descrição a nível de transferência entre registradores (designada em inglês por *Register Transfer Level* ou *RTL*) equivalente, menos abstrata. A descrição RTL é um arquivo textual descrevendo uma interconexão de portas lógicas e flip-flops, independente de dispositivo.

4. Síntese Física

- Processo de mapear a descrição RTL nos elementos do FPGA (no caso da Xilinx e do Foundation. Em outros ambientes, poderia ser para outro tipo de implementação, e.g. um *chip* fabricado sob encomenda).

- As ferramentas básicas são: mapeamento (que decompõe o projeto original em subconjuntos de portas lógicas, com cada subconjunto formado por um conjunto de portas que cabe dentro de exatamente uma LUT), posicionamento (que distribui as CLBs nas LUTs específicas do FPGA, em linhas e colunas da matriz do dispositivo escolhido) roteamento (que conecta as CLBs entre si, por meio das interconexões reconfiguráveis) e geração da imagem binária da configuração do FPGA (um arquivo com denominação <nome do projeto>.bit que contém todos os bits que irão configurar bits da RAM de controle do FPGA).

5. Download

- Processo de descarregar o resultado da síntese física (arquivo .bit) no FPGA.
- Será utilizada uma ferramenta específica do fabricante da placa XS40/XST-1, a empresa Xess (<http://www.xess.com>), não integrada ao ambiente Foundation. Esta ferramenta (XSTools) está instalada nas máquinas do Laboratório. São dois programas: XSLOAD ou GXLOAD (descarrega arquivo .BIT para o FPGA e opcionalmente programa .HEX para o microprocessador 8051 da XS40) e XSPORT ou GXSPORT (comunicação entre PC e placa durante execução).

Parte V - O Ferramental de Hardware a ser Usado

6. Placa XS40

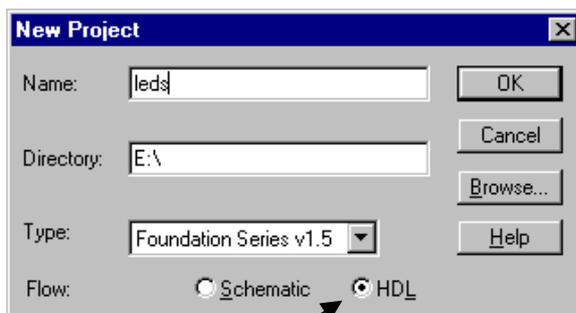
- Placa autônoma de hardware, fabricada pela Xess Corporation, dotada de:
 - um FPGA XC40005XL;
 - um microprocessador 80C51 da Intel;
 - 32Kbytes de RAM e slot para PROM de download;
 - Interface Paralela para conexão ao computador hospedeiro;
 - Conector VGA para capturar sinal de vídeo;
 - 1 Display de 7 segmentos;
 - Vários “jumpers” de configuração, conectores adicionais;
 - regulador de tensão programável (3,3V, 5V, etc).

7. Placa XST-1

- Placa auxiliar à XS40, do mesmo fabricante, contendo:
- Possibilidade de acrescentar mais 64Kbytes de RAM;
- 3 chaves, duas das quais programáveis (entrada de dados e reset do FPGA);
- dip-switch de 8 chaves (entrada de dados);
- 8 leds e 2 displays de 7 segmentos;
- codificador/decodificador estéreo de sinal de som com 2 conectores (in/out);
- conector VGA adicional e conector para teclado PS2;
- área de prototipação para montagens adicionais;
- conector para placa XS40 ou XS95 e conector para todos os sinais da XS40.

Parte VI – A Fazer

1. Recuperar os arquivos “leds.vhd” e “leds.ucf” da homepage da disciplina
2. Abrir o Foundation, iniciando um novo projeto, tendo o cuidado de indicar projeto tipo HDL, pois faremos o projeto de hardware a partir de VHDL.

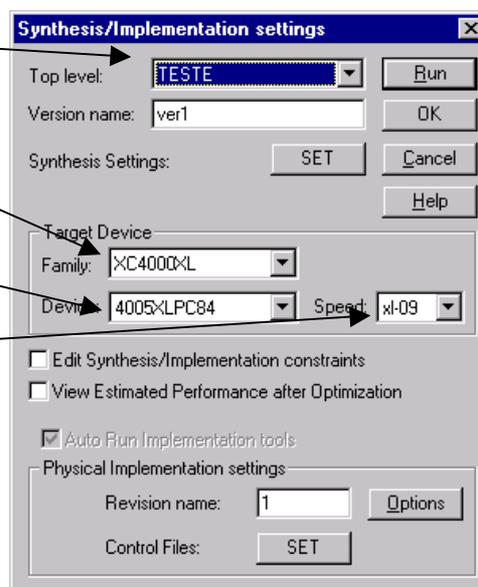


Projeto do tipo HDL

3. Copie os arquivos “leds.vhd” e “leds.ucf” para o diretório do seu projeto, usando a opção de menu Document --> Add... **Atenção:** se o projeto criado não é denominado **leds**, o arquivo ucf copiado deverá ter seu nome alterado para coincidir com o nome do projeto e ter a extensão .ucf. Após inseridos os arquivos, dê duplo clique no arquivo ucf (ver localização na Figura 2), e confira no final do arquivo a relação entre os sinais e os pinos. Deve existir um arquivo com extensão ucf no diretório do projeto, que deve ser substituído pelo arquivo ucf fornecido
4. Insira no projeto o fonte vhdl “leds.vhd”, via menu *Document* → *Add Document*. Selecione o “leds.vhd”. Execute um duplo clique neste arquivo. Estude-o, para compreender sua funcionalidade. **Tarefa 1: Pesquise, Pense e Responda - O que faz o hardware descrito por leds.vhd?**
5. Chamar a ferramenta de síntese lógica automática. Aparecerá o seguinte menu:

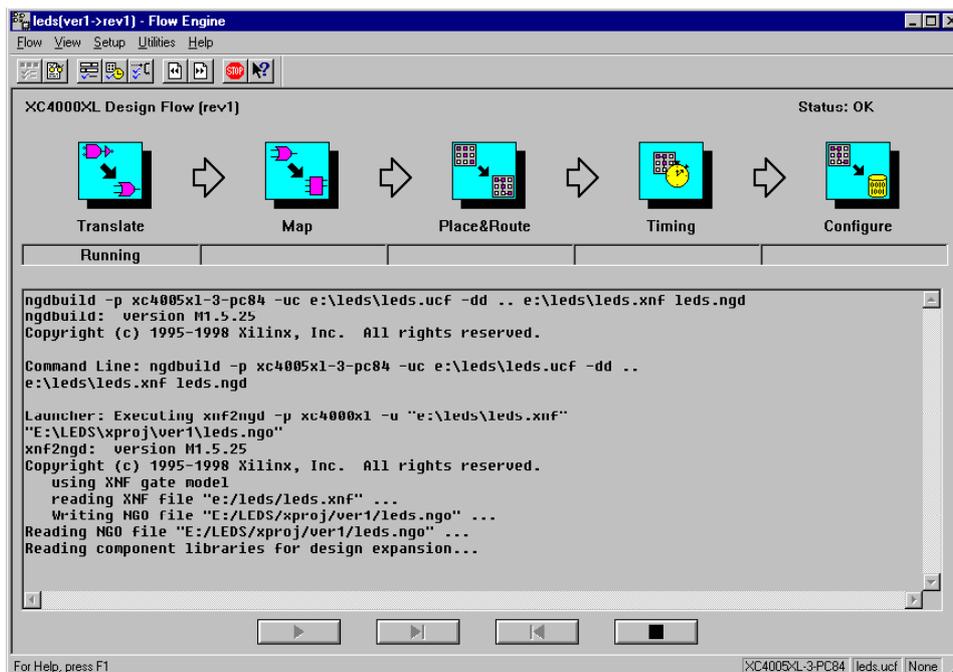
1. Defina a *entity* TESTE como aquela a ser gerada.
ATENÇÃO: é muito importante escolher a entity correta!!

2. Defina a família de FPGAs, no caso é a família XC4000XL (3.3 Volts)
3. Defina o dispositivo da família – 4005 com 84 pinos externos e encapsulamento plástico (PC)
4. Defina a velocidade do dispositivo (deve ser xl – 3)



Execute a síntese lógica (Botão **Run**). A caixa da ferramenta de síntese lógica conterà um “!” ao final da síntese (há um warning apenas).

6. Chamar a síntese física automática. Não há nenhum parâmetro a ser inserido. Aparecerá a janela abaixo, com o progresso da síntese física. Observar as diversas etapas: mapeamento, posicionamento e roteamento, análise de timing (cálculo automático do atraso no caminho mais longo no circuito – crítico) e geração do arquivo de download (extensão **bit**).



7. Agora, analise os relatórios gerados pela tarefas de síntese, seja a síntese lógica (ou independente de tecnologia), seja a síntese física (também chamada de dependente de tecnologia). Existe no software Foundation uma orelha denominada Reports, na mesma sub-janela da janela de fluxo de projeto. Lá existem vários relatórios (ou diretórios de relatórios). **Tarefa 2:** a partir dos relatórios, diga qual a taxa de ocupação dos CLBs do FPGA, e diga qual o tamanho do circuito final, em portas lógicas equivalentes.
8. Observar, por exemplo, que durante a análise de caminho crítico, é indicada uma frequência máxima de funcionamento. **Tarefa 3:** Pesquise, Pense e Responda - Qual foi esta frequência no seu caso? Procure o relatório onde consta esta informação. Descubra e diga também nos relatórios qual é o fio de maior atraso (delay). Qual a relação deste fio com a frequência máxima de operação do circuito? É bom que o valor da frequência máxima seja alto ou baixo? Porquê?

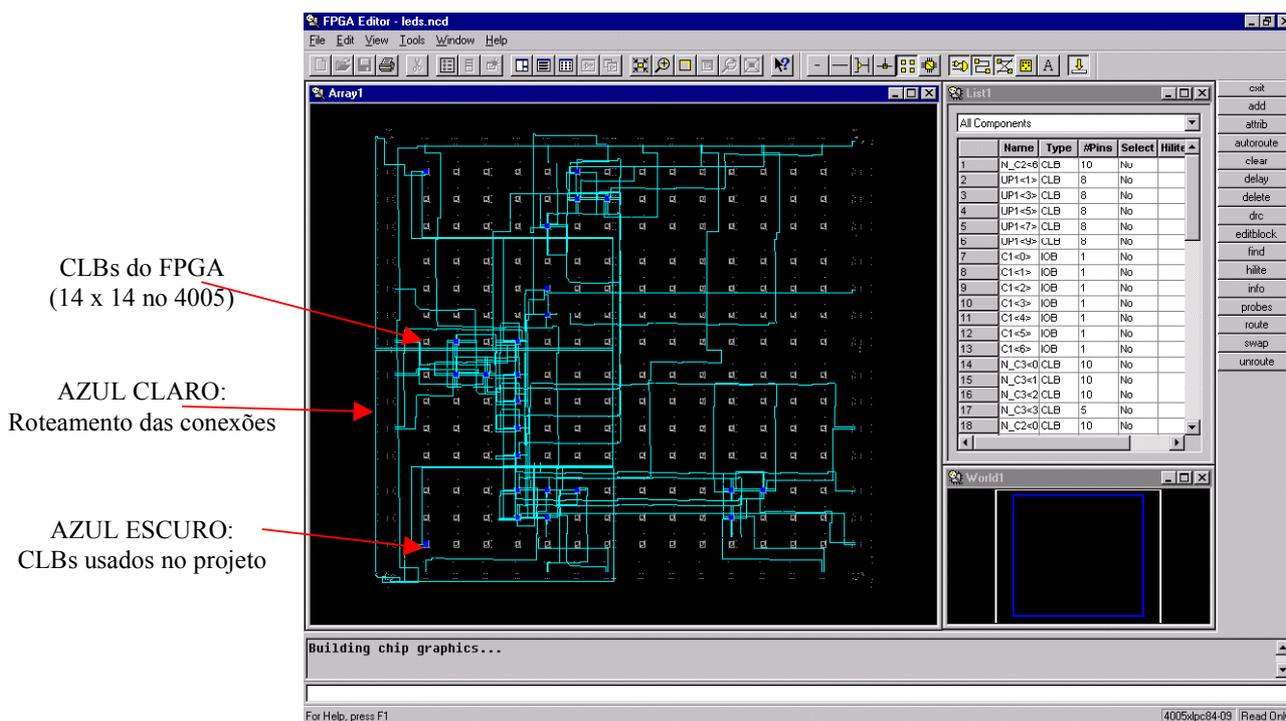


Figura 3 – Representação da planta baixa do circuito leds.pdf implementado internamente no FPGA.

9. Antes de realizar o *download*, observar a arquitetura interna do FPGA. Ir no menu *Tools* → *Implementation* → *FPGA Editor* do Gerenciador de Projetos Foundation. Se aparecer uma janela de diálogo, escolha o arquivo que possui como prefixo o nome do seu projeto. A aparência do circuito mostrado deve ser similar à Figura 3.
10. Tarefa 4: Pesquise, Pense e Responda - Na sua opinião, qual o impacto do número de CLBs na implementação física? Por exemplo, existe limite para o tamanho de circuito implementável?
11. Tarefa 5: Pesquise, Pense e Responda - Qual o impacto das ferramentas de CAD automáticas na implementação física? Por exemplo, observe o “layout” do circuito final obtido com o FPGA Editor. Há muito ou pouco espaço vago? Os fios implementados são sempre os mais curtos ou não? Por quê?
12. Após instalar um dos conjuntos de cabo paralelo, placa e fonte de alimentação no seu computador de trabalho, descarregue o arquivo binário com seu projeto implementado na plataforma de prototipação. Uma das maneiras de fazer isto é abrir uma janela DOS e ir para o diretório onde foram inseridos manualmente os arquivos (item 3 desta lista). Verificar a existência do arquivo *leds.bit*. Digitar no interpretador de comandos DOS **xsload leds.bit**. Outra opção (mais simples, se habilitada) é usar a interface gráfica *gxload* com opção de “drag and drop”. Após realizada esta ação, o circuito deve ser descarregado na placa e deve imediatamente começar a funcionar.

Pronto! Estamos com o algoritmo descrito em VHDL implementado em hardware!

13. Tarefa 6: Altere o hardware implementado para fazer o seguinte: a contagem mostrada nos mostradores de sete segmentos evolui com uma determinada frequência. Pede-se:
 - ❖ Calcule na teoria (a partir das informações no código VHDL) e na prática, qual é esta frequência de contagem (provavelmente diferente, devido à imprecisão do oscilador interno do FPGA. Não é necessário calcular exatamente na versão prática, apenas de forma aproximada, usando, por exemplo um relógio com contagem de segundos;
 - ❖ Altere a frequência de contagem, de forma que esta passe a ser o mais próxima possível de 1Hz;
 - ❖ Inverta a direção aparente de movimento dos leds.

Parte VI - A Fazer e a Entregar

- Projetos original e alterado funcionando, em simulação¹, com test bench. (versão magnética), arquivado (opção de menu Design → Archive Design... do Ambiente Active-HDL), com formas de onda em arquivo;
- Relatório com as respostas às questões aqui colocadas dentro das [Tarefas](#);
- Mostrar o projeto funcionando ao professor até o início da próxima aula.

Percentuais de nota atribuídos às Atividades:

Item Avaliado	Percentual
Projeto completo e simulações	20%
Projeto alterado completo e simulações (Tarefa 6)	50%
Relatório com respostas às questões e tarefas realizadas	30%

¹ ATENÇÃO – Existe necessidade de uma pequena alteração no texto VHDL para que seja possível simular o circuito, pois este usa um componente especial, específico das famílias de dispositivos XC4000. Trata-se do Oscilador interno OSC4. Para poder simular este componente, deve-se acrescentar, na entidade pertinente, a declaração da biblioteca UNISIM da Xilinx. Deve-se igualmente declarar o uso de todos os elementos do package de mesmo nome, UNISIM.