

Plano de projeto para treinamento em microeletrônica

Objetivo:

- Conduzir os profissionais em treinamento por todo o processo de desenvolvimento de um produto em microeletrônica, que seja relevante, desafiador, factível, bem definido e com etapas com crescentes graus de complexidade, em uma metodologia de aprender fazendo (*Learn by Doing*).
- Fabricação?

Metodologia:

- Dividir o desenvolvimento em subprodutos de complexidade crescente, que evoluam para a solução planejada;
- Utilizar principais conceitos de métodos ágeis de desenvolvimento, adaptando-os para cumprimento de requisitos iniciais compulsórios;
- Realizar o fluxo completo de microeletrônica para cada subproduto, promovendo o crescimento de forma recorrente, objetivando fortalecer o entendimento das interdependências entre etapas do fluxo;
- Utilizar ferramenta de rastreo e acompanhamento;
- Utilizar ambiente de desenvolvimento, linguagens, ferramentas e metodologia similar ao utilizado na indústria;

Produto e subprodutos:

P1: Produto final – MP3 player

MP3 player com leitura de cartão SD e saída de áudio em formato I2S para ligar em codec de áudio (DAC);

Blocos: Interface SPI para leitura de cartão SD;
Decodificador MP3 com entrada de bitstream paralelo;
Gerador de clock variável de acordo com taxa de amostragem encontrada no Bitstream;
Interface da saída I2S ou SPDIF;

P1.1: (evolução) Cartão SD com sistema de arquivos e mecanismo de seleção de música;

Blocos adicionais: CPU e memórias;
Barramento;
Firmware;
GPIO – para chaves e LEDs para controle e apresentação;
Modificações: Substituir Interface SPI e decodificador MP3 para barramento;

P1.2: (evolução) LCD gráfico;

Blocos adicionais: Controlador de LCD gráfico;
Modificações: Firmware;

Evolução com subprodutos.

Observação – Para cada subproduto deve-se passar por todo o fluxo de desenvolvimento (alto nível, RTL, verificação e síntese). O Backend só deve ser feito após certos milestones serem

atingidos. Outra observação importante é a teoria de ponto-fixe e sua implementação durante o desenvolvimento (SystemC).

SP1: Produto funcional mínimo – gerador de frequências arbitrárias;

Descrição: Módulo com uma entrada paralela de dados que determina a frequência do sinal de saída. Normalmente o sinal é produzido via acúmulo e overflow e não possui duty-cycle constante.

SP2: Gerador de forma de onda arbitrária;

Descrição: Módulo composto de gerador de frequência arbitrária ligado a um contador, e o contador usado para gerar endereços para uma tabela de busca. A tabela de busca (LUT) pode ser construída com uma memória ou com lógica combinacional. O sinal paralelo de saída pode ser visualizado diretamente ou convertido para valor analógico com um conversor DA. Para testes pode-se usar um conversor DA R-2R. Para cada valor de entrada obtém-se um sinal de determinada frequência. Dessa forma é possível produzir notas musicais.

Obs. Também há a possibilidade de usar um módulo CORDIC para a geração dos valores das amostras, mas fica restrito apenas a forma de onda senoidal. Pode ser bastante interessante constatar o trade-off entre memória e lógica.

SP3: Módulo I2S para uso com chip DAC;

Descrição: A saída analógica pode ser obtida mais facilmente e com mais qualidade com o uso de um conversor AD (codec de áudio). Esse chip já está presente nas placas de FPGA e pode ser facilmente encontrado pronto, como breakout boards.

SP4: Módulo SPDIF/AES3 para uso de interface óptica;

Descrição: A saída analógica também pode ser obtida com facilidade através do uso de um LED como saída para ser acoplado via fibra óptica em um conversor SPDIF toslink, facilmente encontrado comercialmente. Nesse caso o sinal de dado e de clock são enviados por uma única saída e devem ser combinados usando *Differential Manchester encoding (biphase mark coding)*.

SP5: Módulo ADPCM;

Descrição: Descompressor ADPCM (Adaptative Differential Pulse Code Modulation). Trata-se de um algoritmo muito simples. Há várias versões, sendo o IMA ADPCM bastante compatível, com codecs de software no Windows e no Linux. A compressão é fixa em 4:1. Inicialmente as amostras comprimidas podem ser guardadas em uma memória ROM.

SP6: Módulo MP2;

Descrição: Descompressor MP2 ([MPEG-1 Audio Layer II - Wikipedia](#)). Por simplicidade apenas uma taxa de amostragem um bitrate deve ser suportado inicialmente. Alguns blocos precisam ser desenvolvidos nessa etapa –

- a) decodificador de código Huffman;
- b) Separador de campos de bitstream;
- c) Aplicação dos fatores de escala;
- d) IMDCT (Transformada Inversa DCT Modificada).

Os bits de entrada podem ser armazenados em ROM.

SP7: Módulo leitor cartão SD;

Descrição: Módulo SPI master ligado a um cartão SD; O bitstream pode ser guardado diretamente no cartão, sem formatação nenhuma. Outra opção é o uso de memória flash SPI. Esse módulo pode substituir as memórias ROM nos subprodutos anteriores. É possível evoluir para utilizar a interface de 4 bits do cartão, que equivale a quad-SPI.

SP8: Módulo decodificador MP3;

Descrição: Acréscimo dos módulos adicionais para decodificar MP3. Em particular temos a IMDCT de dois comprimentos diferentes, o bitstream mais complexo, o tratamento da transição entre blocos curtos e longos.

SP9: Processador e barramento;

Descrição: desenvolvimento de um SoC (System-on-Chip) bem básico, formado de processador, memória ROM de instrução, memória RAM, e interface de barramento. Nesse subproduto é suficiente a implementação de uma interface GPIO simples.

SP10: Conversão dos módulos dedicados em aceleradores;

Descrição: Mudança das interfaces dos módulos de decodificador MP2/MP3, leitor de cartão SD e (talvez) saída I2S/SPDIF para barramento intrachip.

SP11: Controlador de LCD;

Descrição: Controlador de LCD gráfico. Há o padrão MIPI DSI que pode ser investigado. Os módulos LCD pequenos possuem geralmente interface SPI ou paralela (específica). O módulo deve ter interface com o barramento do chip.