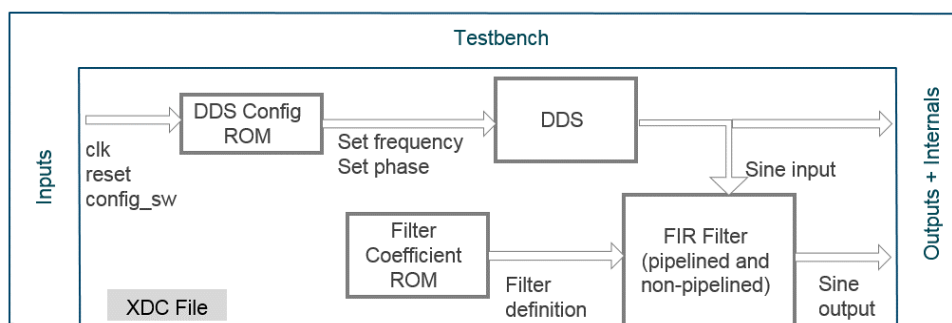


## Projeto 1

A cada grupo de estudantes será atribuído um circuito diferente de processamento de sinais, descrito numa linguagem de descrição de *hardware*. O circuito integra um sintetizador digital direto (DDS) e um filtro FIR de ordem 7. O filtro é fornecido com duas arquiteturas, com e sem *pipelining* - para considerar uma ou outra, altere a arquitetura na declaração do filtro (fir8\_test.vhd, linha 66) entre **rtl** ou **rtl\_piped**. Cada um dos módulos (DDS e FIR) é acompanhado de uma memória que fornece parâmetros de configuração. No caso da DDS, a frequência e a fase inicial da senoide são disponibilizadas em vetores de 32 bits. No caso do filtro, os 8 coeficientes são fornecidos com a quantização adequada. É fornecido também um ficheiro de testbench e um ficheiro com restrições temporais. A figura 1 representa a arquitetura do circuito e o conjunto de ficheiros fornecidos.



**Figura 1 – Ficheiros fornecidos / Set of files provided**

Seguindo a mesma metodologia dos trabalhos laboratoriais, os estudantes deverão concretizar seguintes passos:

1. Analisar a arquitetura do circuito RTL (após a fase *Elaborate*).
2. Analisar os coeficientes do filtro e identificar a sua função de transferência (pode usar o Matlab para o efeito, por ex. a função **freqz**).
3. Simular o circuito usando o simulador do Vivado e explicar o seu funcionamento com base nos resultados de simulação. As formas de onda devem ser incluídas no relatório final e visualizadas todos os sinais necessários para explicar o funcionamento dos diferentes módulos do circuito.
4. Repetir o ponto anterior com novos coeficientes no filtro, de forma a implementar a função de transferência proposta para grupo de estudantes. Devem também ser alterados os parâmetros de configuração da DDS para que seja possível observar a sua resposta nos resultados de simulação.
5. Sintetizar o projeto usando diferentes configurações e observar os efeitos das opções tomadas. Observar e comentar o efeito das opções “*flatten hierarchy*” e “*max\_dsp*” tanto no desempenho como na área ocupada.
6. Identificar os caminhos críticos, encontrar a velocidade máxima de relógio, e comentar o efeito que um filtro com *pipelining* tem no desempenho global do circuito.

7. Implementar o projeto usando diferentes opções e comparar os resultados/relatórios obtidos com os que resultam no final do processo de síntese;
8. Utilizar a ferramenta IP Packager para gerar um IP do circuito fornecido.
9. Utilizar a ferramenta IP Integrator para construir um circuito que integre, pelo menos, o circuito fornecido e um módulo IP da biblioteca do Vivado (será atribuído um módulo diferente a cada grupo).
10. Simular o circuito global usando o simulador do Vivado e explicar o seu funcionamento com base nos resultados de simulação.

**O relatório deverá ser elaborado de acordo com o modelo fornecido!**

## Project 1

---

Each student group will be assigned a different signal processing circuit, described in a hardware description language. The circuit includes a direct digital synthesizer (DDS) and an FIR filter of order 7. The filter is provided with two architectures, with and without pipelining - to consider one or the other, change the architecture definition in the component declaration (fir8\_test.vhd, line 66). Each of the modules (DDS and FIR) is accompanied by a memory block that provides configuration parameters. In the case of DDS, the frequency and the initial phase of the sinusoid are available in 32-bit vectors. In the case of the filter, the 8 coefficients are provided with the appropriate quantization. Testbench and XDC files (time restrictions) are also provided. Figure 1 (see Portuguese version) represents the set of files provided and circuit's architecture.

Using the same methodology of the work proposed in the laboratory assignments, students should follow the steps below:

1. Elaborate the design and analyze the architecture.
2. Analyze the filter coefficients and identify its transfer function (you can use Matlab for this purpose, e.g. the **freqz** function).
3. Simulate the circuit using the Vivado simulator and explain its operation based on the simulation results. Waveforms must be included in the final report and all possible signals visualized in order to explain the functionality of each circuit's modules.
4. Repeat the previous task with new coefficients in the filter, in order to implement the transfer function proposed for your group. The DDS configuration parameters must also be changed so that it is possible to observe the filter's response in the simulation results.
5. Synthesize the design using different settings and observe the effects of the options taken. Observe and comment the effect of the "*flatten hierarchy*" and "*max\_dsp*" options in both performance and area (as minimum goals).
6. Identify critical paths, find the maximum clock frequency and comment the effect of filter pipelining in the overall circuit performance.
7. Implement the design using different options and compare the results/reports obtained with those at the end of the synthesis process;
8. Use the IP Packager tool to generate an IP of the circuit provided.
9. Use the IP Integrator tool to build a circuit that integrates your circuit with an IP module from Vivado library (a different module will be assigned to each group).
10. Simulate the global circuit using the Vivado simulator and explain how it works using simulation waveforms.

**The report must be prepared according to the template provided!**