# 1. Estrutura do Testbench

O testbench foi desenvolvido para verificar a correta operação do módulo controladorCriptografia, responsável por orquestrar as etapas do algoritmo de criptografia AES-128.

Ele contém os seguintes blocos:  
- Declaração de sinais: sinais de clock, reset, controle, entrada e saída.  
- Instanciação do DUT (Design Under Test).  
- Gerador de clock.  
- Task de verificação de cifragem (executarTeste).  
- Monitoramento do estado interno da FSM.

# 2. Sinais Principais

- clk: Sinal de clock principal, gerado por um módulo auxiliar.  
- rst: Reset síncrono para inicialização da FSM.  
- start: Início do processo de criptografia.  
- estado: Estado atual da máquina de estados.  
- chave: Chave de 128 bits para criptografia.  
- palavra: Palavra de 128 bits a ser criptografada.  
- cifra: Resultado da operação de criptografia.  
- done: Indica conclusão do processo.  
- Saídas intermediárias (auxEstadoEntrada, auxChaveEntrada, etc.): Usadas para debug e análise interna dos passos do algoritmo AES.

# 3. Gerador de Clock

O módulo clockGenerator é instanciado com período de 10ns (100 MHz).

```verilog  
module clockGenerator #(parameter period = 5)(output clk);  
 reg outClk = 1'b0;  
 always begin  
 #(period/2) outClk = ~outClk;  
 end  
 assign clk = outClk;  
endmodule  
```

# 4. Task executarTeste

A task ***executarTeste*** é responsável por executar um ciclo completo de teste sobre o módulo *controladorCriptografia*, realizando a criptografia de uma palavra de 128 bits com uma chave de 128 bits e verificando se a saída corresponde ao valor esperado.

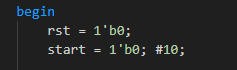
Parâmetros da task



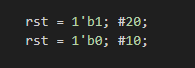
*id*: Identificador numérico do teste. Serve para referenciar e exibir qual teste está sendo executado no console de simulação.  
- *esperado*: Valor de 128 bits que representa a cifra esperada para o par (palavra, chave) definido anteriormente.

Sequência de Operações Executadas  
----------------------------------

1. Inicialização de Sinais:  
Os sinais de controle são zerados para garantir estado estável antes de aplicar o reset. Um pequeno atraso é inserido.



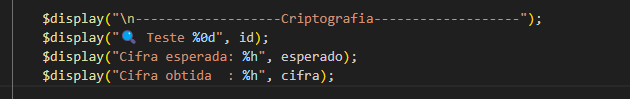
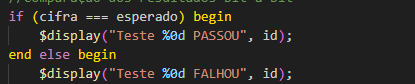
2. Reset do Módulo:  
Reset ativado por 20 unidades de tempo para reinicializar a FSM, e desativado em seguida.



3. Acionamento do Start:  
O sinal `start` é ativado por 15 unidades de tempo e logo em seguida desativado.



4. Aguardando Finalização:  
  
Espera até que `done` seja ativado, sinalizando que o DUT concluiu a criptografia.

5. Exibição de Resultados:  
6. Verificação da Cifra:  


# 5. Testes Utilizados

### 5.1. Testes Baseados em FIPS-197

**Teste 0 (Entrada** 128'h00112233445566778899aabbccddeeff**)**:

* + **Entrada (Plaintext)**: 128'h00112233445566778899aabbccddeeff
  + **Chave (Key)**: 128'h000102030405060708090a0b0c0d0e0f
  + **Cifra (Ciphertext)**: 128'h69c4e0d86a7b0430d8cdb78070b4c55a

Esses valores são diretamente extraídos dos **vetores de teste do FIPS-197** para **AES-128**. O vetor de teste com **entrada** 00112233445566778899aabbccddeeff e **chave** 000102030405060708090a0b0c0d0e0f gera o **ciphertext** esperado 69c4e0d86a7b0430d8cdb78070b4c55a.

**Teste 1 (Entrada** 128'h3243f6a8885a308d313198a2e0370734**)**:

* + **Entrada (Plaintext)**: 128'h3243f6a8885a308d313198a2e0370734
  + **Chave (Key)**: 128'h2b7e151628aed2a6abf7158809cf4f3c
  + **Cifra (Ciphertext)**: 128'h3925841d02dc09fbdc118597196a0b32

Esses valores também são parte dos **vetores de teste do FIPS-197**. O **plaintext** 3243f6a8885a308d313198a2e0370734 criptografado com a chave 2b7e151628aed2a6abf7158809cf4f3c resulta no **ciphertext** 3925841d02dc09fbdc118597196a0b32.

**Teste 2 (Entrada** 128'h00000000000000000000000000000000**)**:

* + **Entrada (Plaintext)**: 128'h00000000000000000000000000000000
  + **Chave (Key)**: 128'h2b7e151628aed2a6abf7158809cf4f3c
  + **Cifra (Ciphertext)**: 128'h66e94bd4ef8a2c3b884cfa59ca342b2e

A entrada **nula** (com todos os bits igual a zero) também é um caso de teste definido pelo **FIPS-197**. O resultado esperado para a criptografia de dados nulos com a chave especificada é 66e94bd4ef8a2c3b884cfa59ca342b2e.

**Teste 3 (Entrada** 128'hffffffffffffffffffffffffffffffff**)**:

* + **Entrada (Plaintext)**: 128'hffffffffffffffffffffffffffffffff
  + **Chave (Key)**: 128'h2b7e151628aed2a6abf7158809cf4f3c
  + **Cifra (Ciphertext)**: 128'h3f5b8cc9ea855a0afa7347d23e8d664e

Testes com **dados máximos** (todos os bits definidos como 1) também fazem parte dos testes típicos de verificação de segurança e funcionamento do algoritmo. O resultado da criptografia desses dados nulos com a chave fornecida gera o **ciphertext** 3f5b8cc9ea855a0afa7347d23e8d664e.

### 5.1. Teste Não Baseados em FIPS-197

**Teste 4 (Entrada Arbitrária** 128'haabbccddeeff00112233445566778899**)**:

* + **Entrada (Plaintext)**: 128'haabbccddeeff00112233445566778899
  + **Chave (Key)**: 128'h102030405060708090a0b0c0d0e0f000
  + **Cifra (Ciphertext)**: 128'h2be52b98821c28a467897944fa4ac1bc

Este **teste não é baseado diretamente nos vetores de teste do FIPS-197**, mas utiliza **dados arbitrários** para testar o AES-128. Esse tipo de teste é útil para verificar se o algoritmo lida corretamente com entradas variadas e não segue um padrão conhecido. Embora os valores esperados para esse teste sejam definidos, eles não fazem parte dos vetores oficiais fornecidos pelo FIPS-197.   
  
Todos os valores utilizados foram verificados tanto no documento FIPS-197 quanto nas calculadoras online:

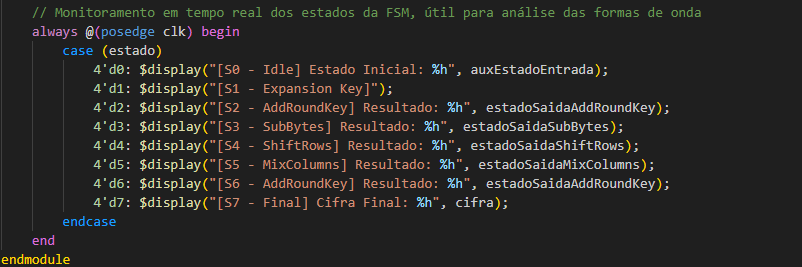
* <https://testprotect.com/appendix/AEScalc>
* <https://csrc.nist.gov/pubs/fips/197/final>
* <https://xor.pw/>

Foram utilizados 5 testes com valores provenientes de fontes confiáveis:

- Teste 0: AES padrão com entrada incremental (https://testprotect.com)  
- Teste 1: Vetor oficial FIPS-197 (página 33 do documento)  
- Teste 2: Palavra e chave com todos os bits zerados (https://testprotect.com)  
- Teste 3: Palavra com bits 1 e chave zero (https://testprotect.com)  
- Teste 4: Vetor genérico com salto regular (https://testprotect.com)

# 6. Monitoramento de Estados

Um bloco always acompanha a FSM do módulo controladorCriptografia, exibindo o estado atual e as saídas intermediárias relacionadas.



Isso facilita a depuração via formas de onda e correlaciona os estágios do algoritmo com seus resultados parciais.

# 7. Conclusão

# A task *executarTeste* realiza a validação funcional do módulo de criptografia AES-128, iniciando com a aplicação de um reset seguido do sinal de start para disparar o processamento da cifra; em seguida, aguarda o sinal *done* que indica a conclusão da operação e, então, compara o valor cifrado gerado com o valor esperado fornecido como parâmetro, exibindo no console se o teste foi aprovado ou falhou. Paralelamente, um bloco *always* monitora em tempo real os estados da máquina de estados (FSM), imprimindo os resultados intermediários de cada fase do algoritmo AES (como *AddRoundKey, SubBytes, ShiftRows e MixColumns*), facilitando a depuração por meio de formas de onda e conferência textual.