

Controle de exibição de 7 segmentos Nexys-4

Recomendações e boas práticas de design

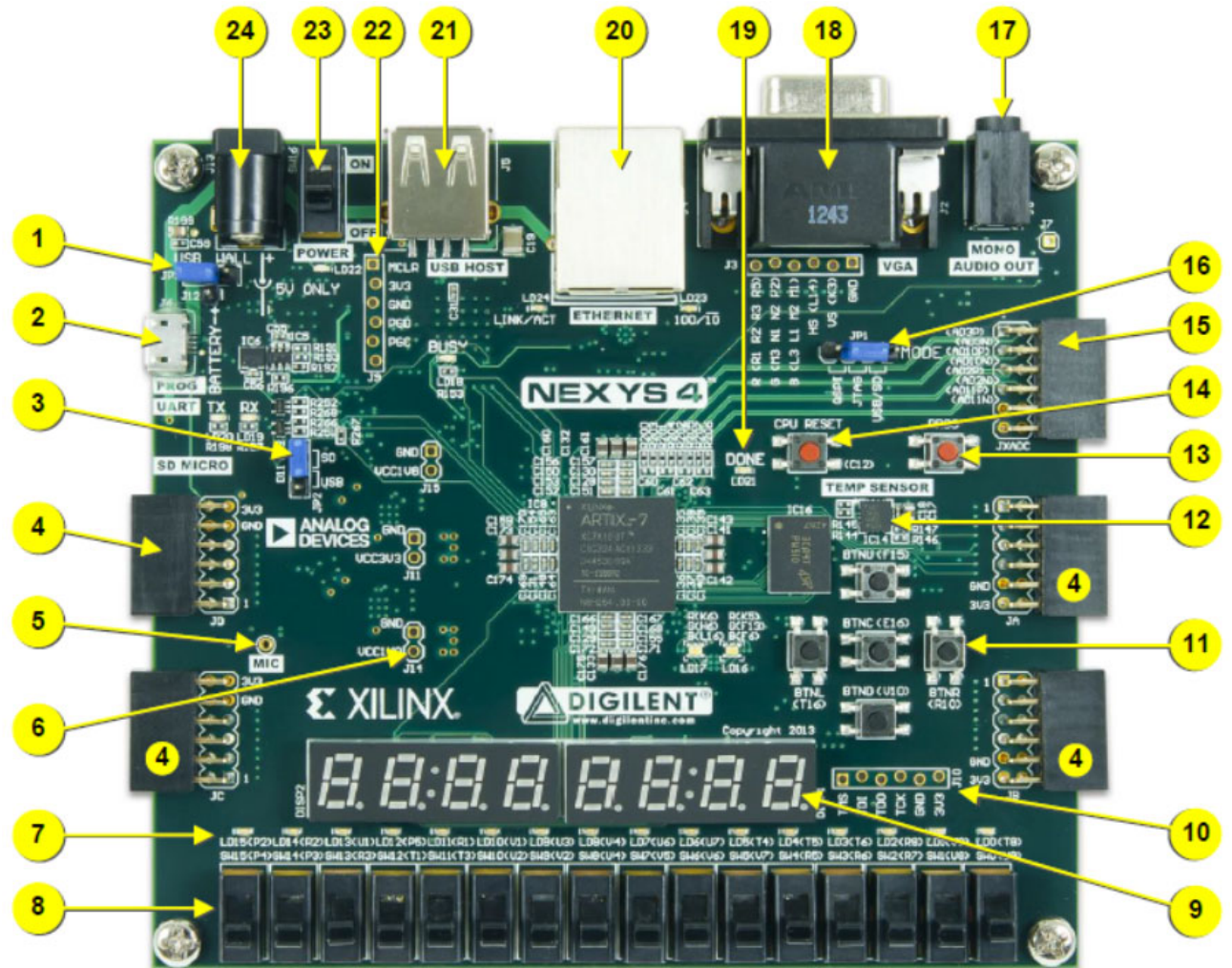
Especificação de restrições e análise de tempo

AULA 3

IOUL IIA SKL I AROVA

Placa de Desenvolvimento Nexys-4

- Monitores de 7 segmentos
- ...



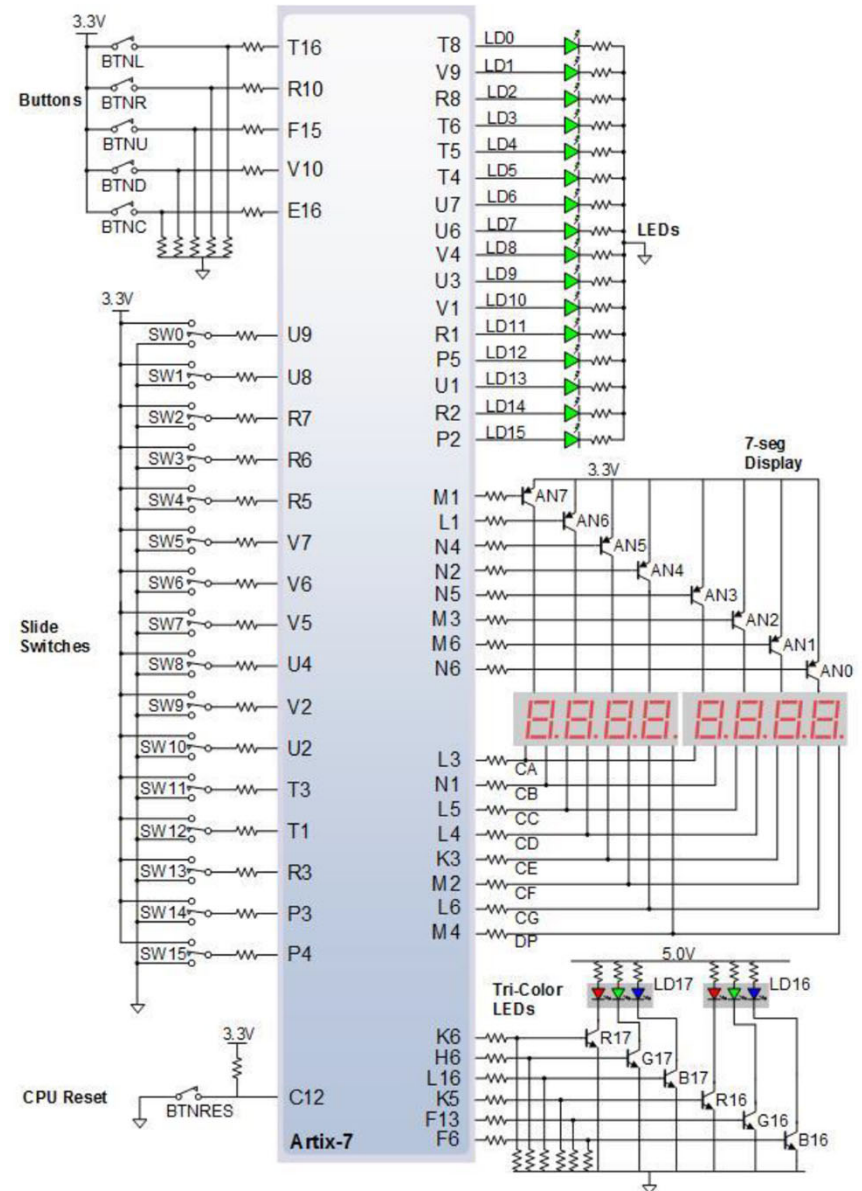
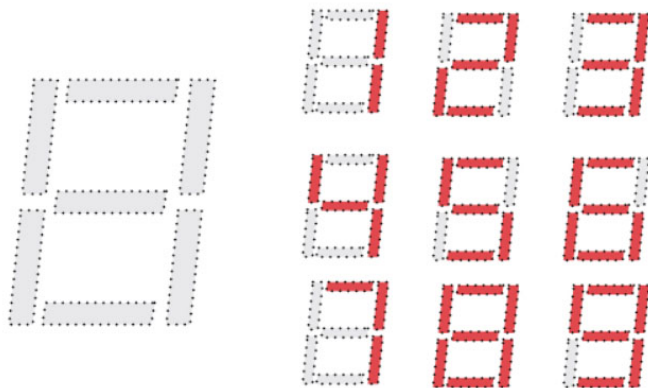
FPGA: xc7a100Tcsg324-1

Monitores de 7 segmentos

A placa Nexys-4 contém dois displays LED de sete segmentos com ânodo comum de quatro dígitos, configurados para se comportar como um único display de oito dígitos.

Cada um dos oito dígitos é composto por sete segmentos dispostos em um padrão de “figura 8”, com um LED embutido em cada segmento.

Os LEDs de segmento podem ser iluminados individualmente, de modo que qualquer um dos 128 padrões pode ser exibido em um dígito, iluminando determinados segmentos de LED e deixando os outros escuros.



Monitores de 7 segmentos

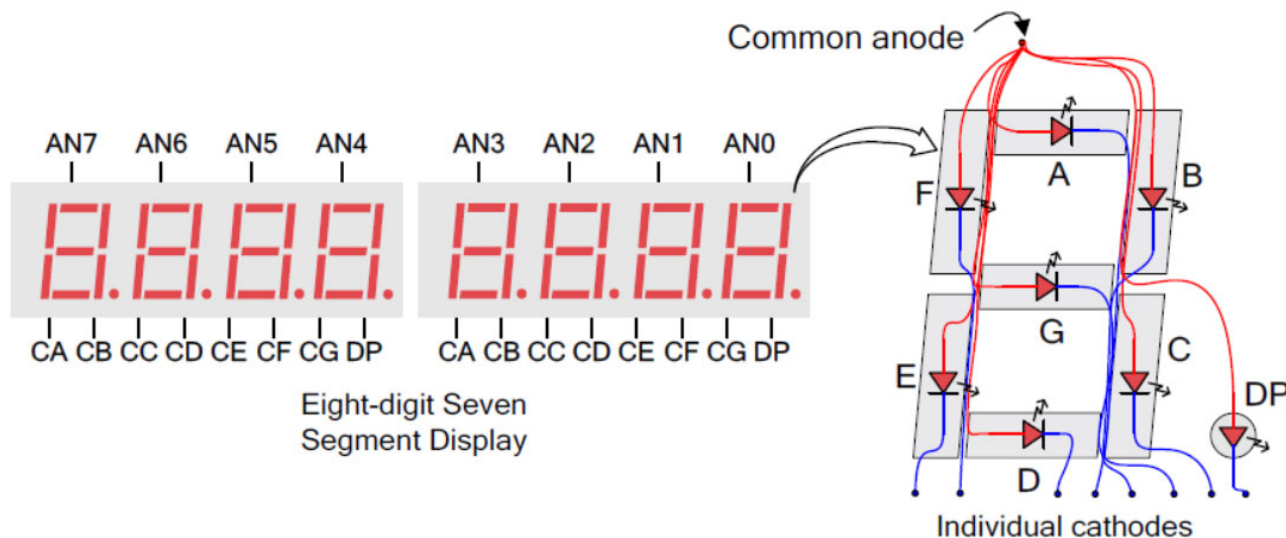
Os ânodos dos sete LEDs que formam cada dígito são interligados em um nó do circuito “ânodo comum”, mas os cátodos dos LED permanecem separados.

Os sinais de ânodo comum estão disponíveis como oito sinais de entrada de “habilitação de dígitos” para o display de 8 dígitos.

Os cátodos de segmentos semelhantes em todos os quatro monitores são conectados em sete nós de circuito rotulados de CA a CG.

Este esquema de conexão de sinal cria um display multiplexado, onde os sinais do cátodo são comuns a todos os dígitos, mas só podem iluminar os segmentos do dígito cujo sinal do ânodo correspondente é afirmado.

Ambos os sinais AN0..7 e CA..G/DP são reduzidos quando ativos.



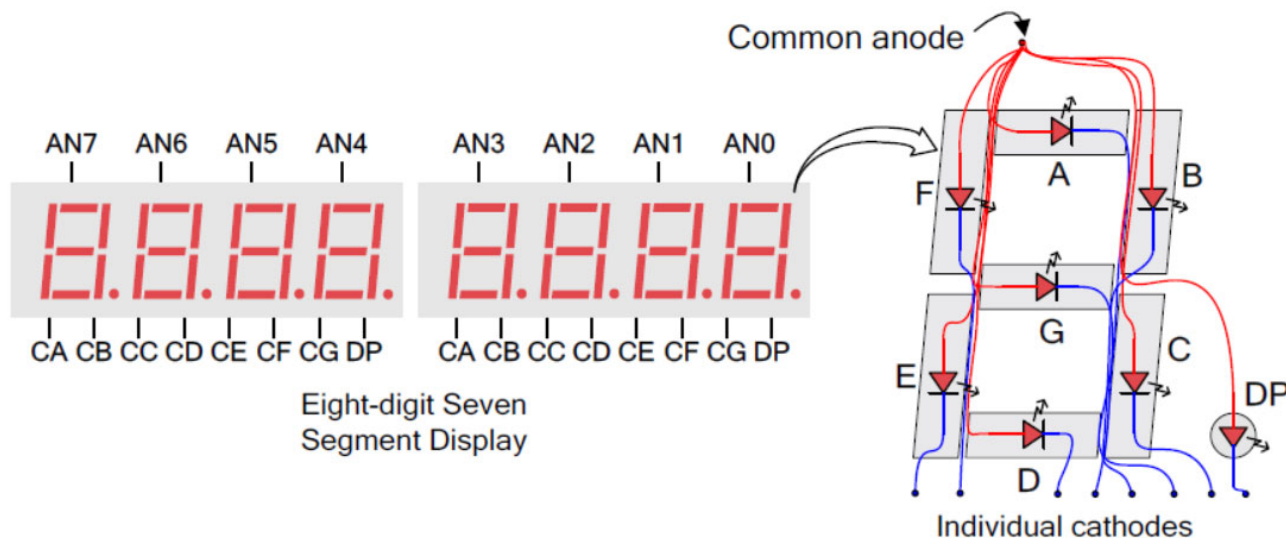
Controlador de displays de 7 segmentos

Um circuito controlador de display de varredura pode ser usado para mostrar um número de oito dígitos neste display.

Este circuito aciona os sinais anódicos e os padrões catódicos correspondentes de cada dígito em uma sucessão repetida e contínua a uma taxa de atualização mais rápida do que o olho humano pode detectar.

Cada dígito é iluminado apenas um oitavo do tempo, mas como o olho não consegue perceber o escurecimento de um dígito antes de ser iluminado novamente, o dígito parece continuamente iluminado.

Se a taxa de atualização, ou “atualização”, for reduzida para cerca de 45Hz, uma cintilação pode ser notada no display.

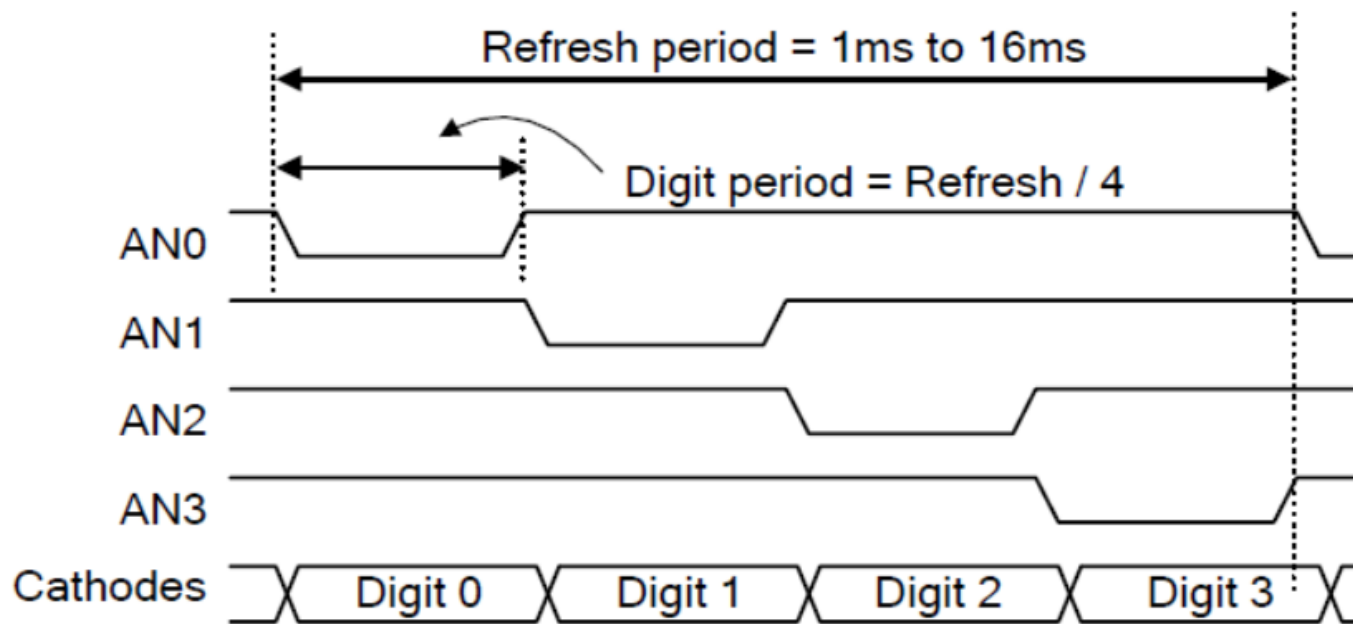


Controlador de displays de 7 segmentos

Para que cada um dos oito dígitos pareça brilhante e continuamente iluminado, todos os oito dígitos devem ser acionados uma vez a cada 1 a 16 ms, para uma frequência de atualização de cerca de 1 KHz a 60 Hz.

Por exemplo, em um esquema de atualização de 62,5 Hz, todo o display seria atualizado uma vez a cada 16 ms e cada dígito seria iluminado durante 1/8 do ciclo de atualização, ou 2 ms.

O controlador deve abaixar os cátodos com o padrão correto quando o sinal do ânodo correspondente for abaixado.



Controlador de displays de 7 segmentos

Sugestões de projeto:

- você tem que alternar entre os dígitos devagar o suficiente para que nossos olhos humanos vejam a luz
- ... mas rápido o suficiente para que não haja oscilação
- aplique um clock de 800 Hz (período de clock = 1,25 ms)
- isso significa que a taxa de atualização de todos os 8 dígitos é de 100 Hz e cada dígito é iluminado por 1,25 ms em um período de 10 ms.

Sinais Assíncronos

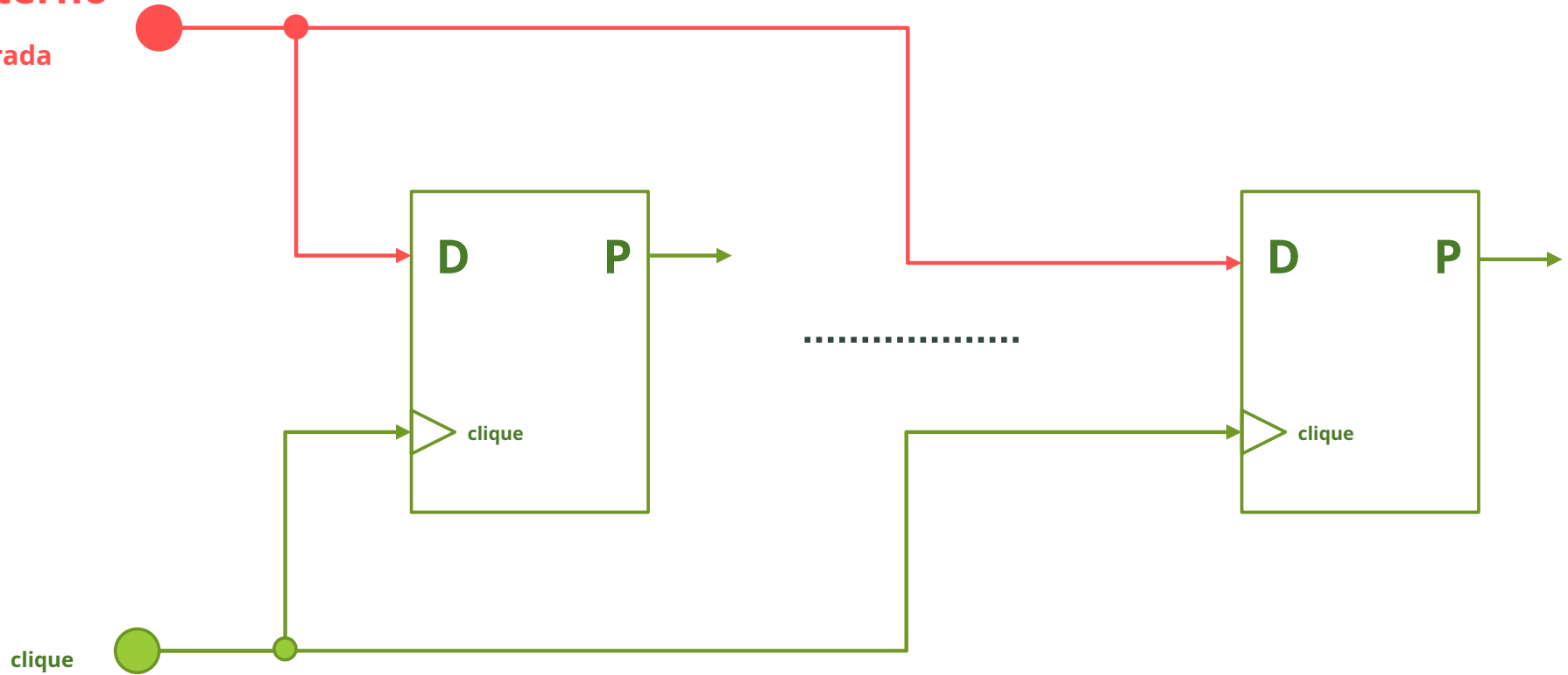
Os circuitos cronometrados são **síncrono**

Circuitos ou sinais sem clock são circuitos ou sinais são **assíncrono** Os circuitos síncronos possuem entradas assíncronas:

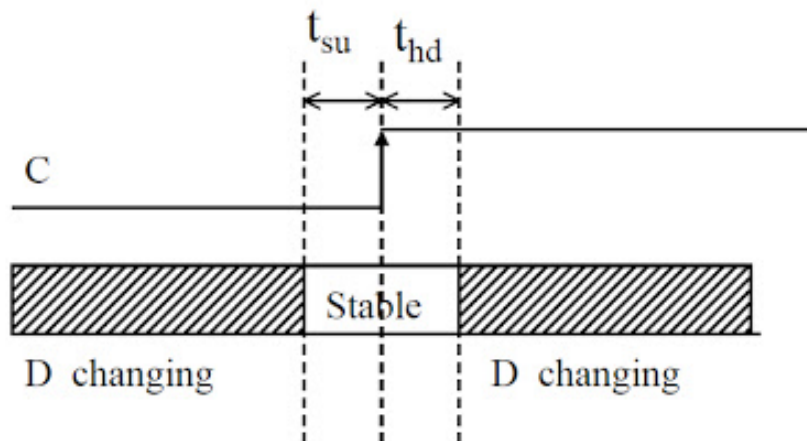
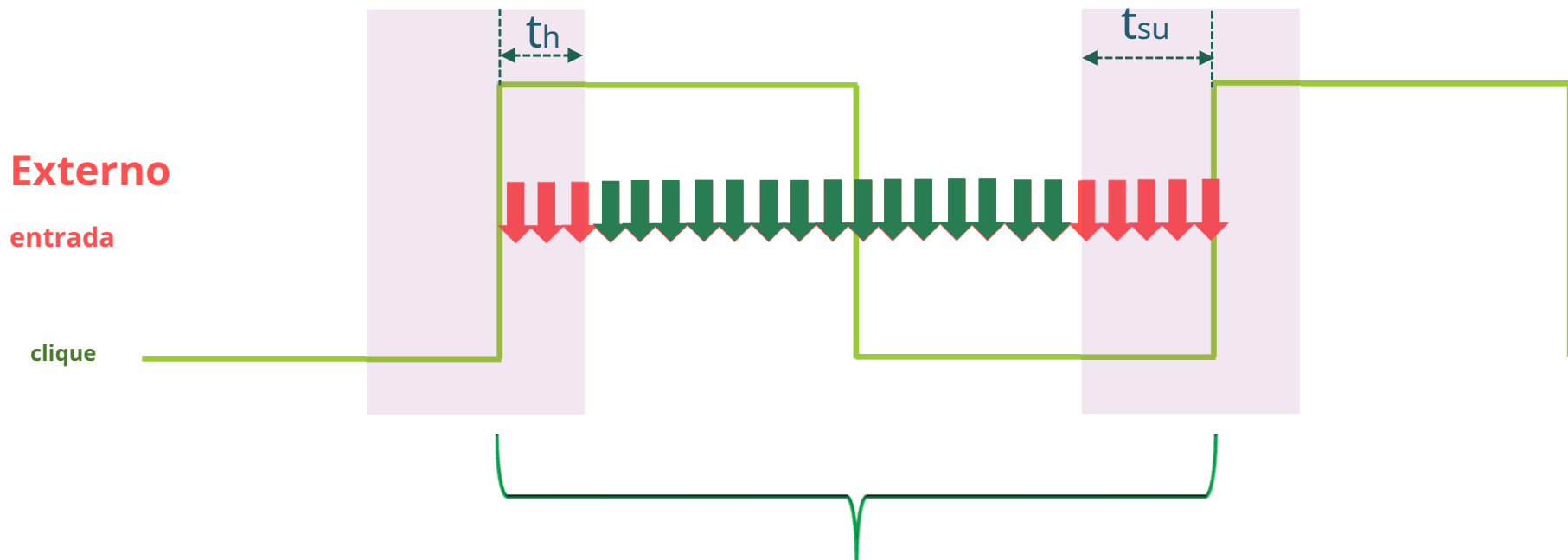
- Sinal de reset, espera de memória, entrada do usuário, etc.
- As entradas podem mudar a qualquer momento
- Devemos sincronizar a entrada com o nosso relógio
- As entradas podem violar os tempos de configuração/manutenção do flip-flop

Entradas assíncronas

Externo
entrada



Sinais Assíncronos



Problema de metaestabilidade:

- Ocorre quando a entrada muda perto da borda do clock ativa
- Nem toda transição de sinal que viola o t_{su} ou t_h resulta em uma saída metaestável
- A probabilidade de um registro entrar em um estado metaestável e o tempo necessário para retornar a um estado estável variam dependendo da tecnologia de processo usada para fabricar o dispositivo e das condições operacionais.
- Na maioria dos casos, os registradores retornarão rapidamente a um estado definido estável

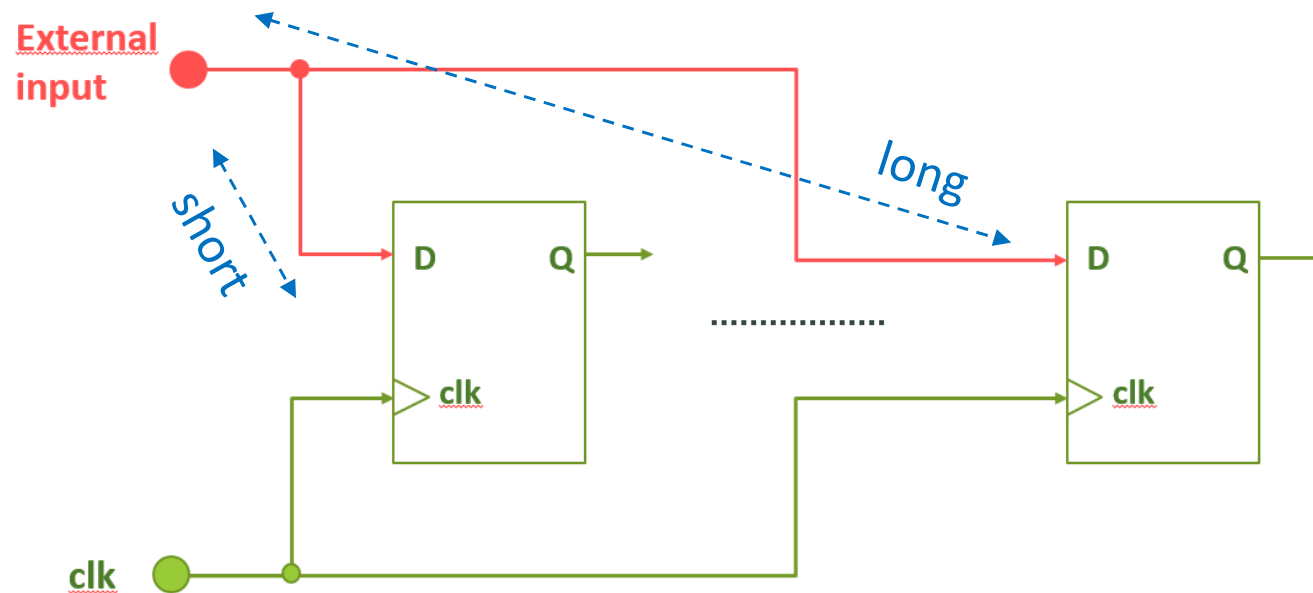
Entradas assíncronas

Além de **metaestabilidade**:

Pequenas diferenças de atraso significam que os registradores podem discordar no valor de entrada

“Problema de valor inconsistente”

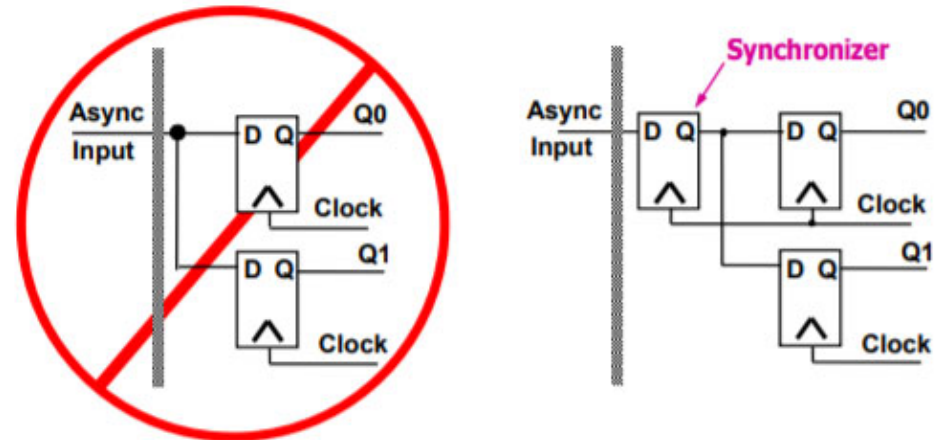
- Dois caminhos da entrada para dois registros diferentes:



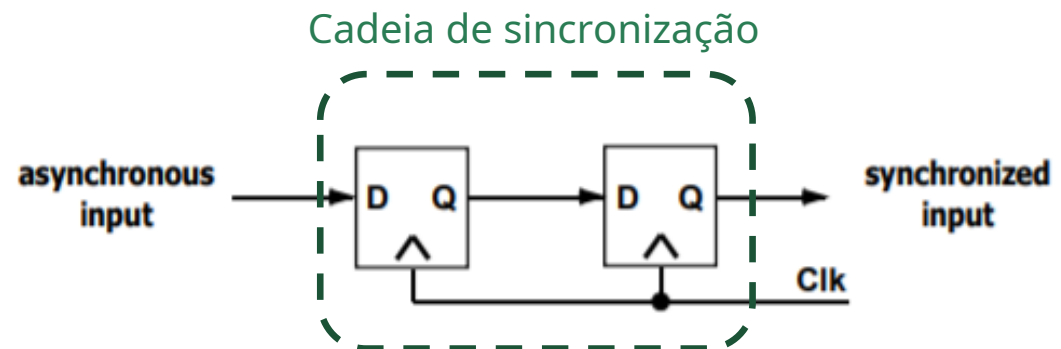
Tratamento de entradas assíncronas

Nunca espalhe entradas assíncronas

- Sincronizar no limite do circuito – **registro de sincronização**
- Sinal sincronizado de fan-out
- A probabilidade de falha nunca pode ser zero



- Para minimizar as falhas devido à metaestabilidade nas transferências de sinal assíncronas, coloque em cascata dois (ou mais) flip-flops
 - Sincroniza efetivamente duas vezes
 - Ambos teriam que falhar para que o sistema falhasse



Análise de tempo

Análise de tempo é a análise metódica de um circuito digital para determinar se **restrições de tempo** impostas por componentes ou interfaces sejam atendidas. Normalmente, isso significa que você está tentando provar que todos os tempos de configuração, retenção e largura de pulso estão sendo atendidos.

Análise de tempo pode ser **estático** ou **dinâmico**.

- **Análise de tempo dinâmico (DTA)** verifica a funcionalidade do design aplicando vetores de entrada e verificando os vetores de saída corretos. Esta abordagem é uma extensão da simulação e garante que a temporização do circuito seja testada em seu contexto funcional.
- **Análise de tempo estático (STA)** verifica os requisitos de atraso estático do circuito sem quaisquer vetores de entrada ou saída.

O DTA deve ser realizado e a funcionalidade do projeto deve ser liberada antes que o projeto seja submetido ao STA.

DTA e STA não são alternativas entre si.

A qualidade do DTA aumenta com o aumento dos vetores de teste de entrada. O

DTA pode ser usado tanto para projetos síncronos quanto assíncronos.

O STA não pode ser executado em projetos assíncronos e, portanto, o DTA é a melhor maneira de analisar projetos assíncronos.

O DTA também é mais adequado para projetos com relógios que cruzam vários domínios.

Para que serve a análise de tempo?

Questões

- Qual é o atraso mais longo no seu circuito?
- O sistema pode operar em uma frequência de clock alvo?
- Os requisitos de tempo são atendidos para todos os caminhos do seu projeto?
- Como determinar a frequência máxima de operação de um sistema?

Análise de tempo estático

Análise de tempo estático é um método de validação do desempenho temporal de um projeto, verificando **todos os caminhos possíveis** por violações de tempo.

Considera o “pior” atraso possível em cada elemento lógico, mas não a operação lógica do circuito.

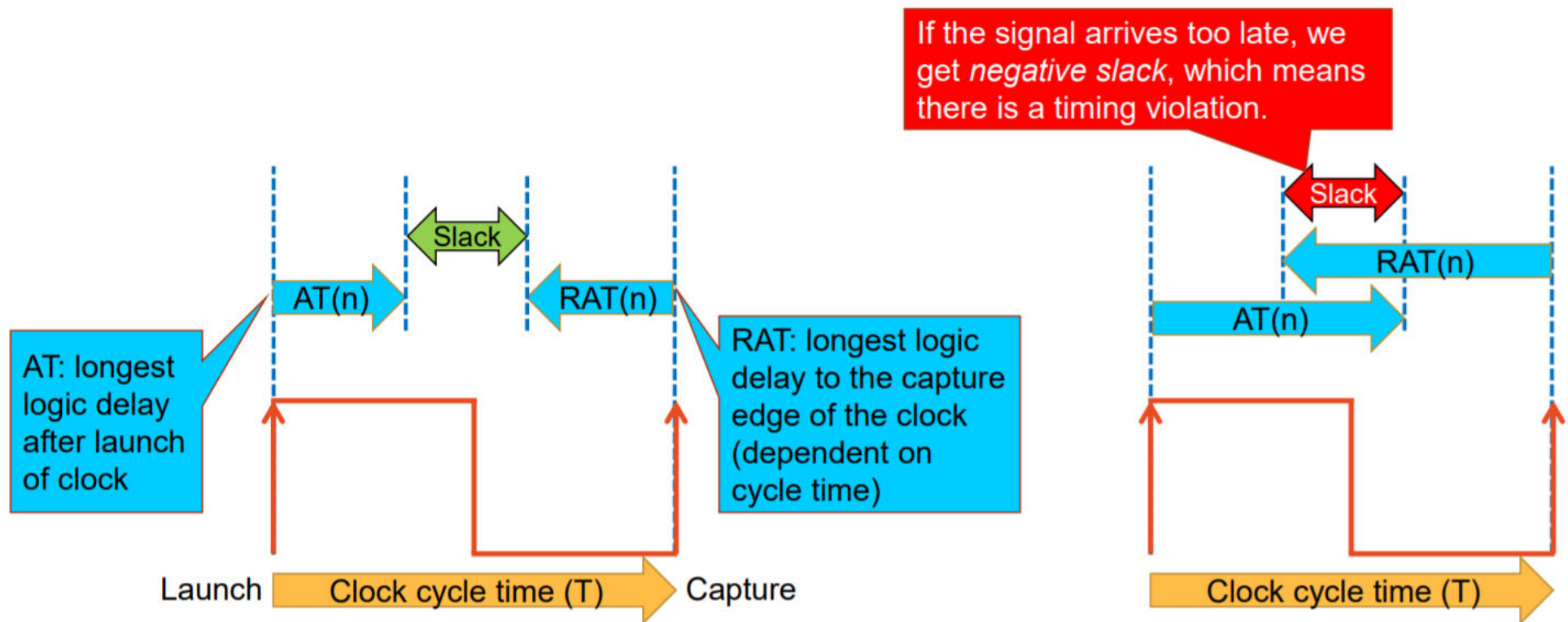
Em comparação com a simulação de circuito, a análise de temporização estática é mais rápida e completa.

Entretanto, a análise de temporização estática verifica o projeto apenas quanto à temporização adequada, e não quanto à funcionalidade lógica correta.

A folga

Folga positiva = OK

Folga negativa = problema: **violação de tempo**



$$*AT=AAT$$

Restrições de tempo

Como a ferramenta STA sabe qual é o período de clock necessário?

Temos que contar!

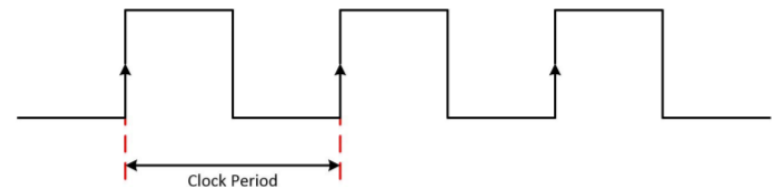
Temos que definir restrições para o design.

Isso geralmente é feito usando a sintaxe Synopsys Design Constraints (SDC).

No Vivado, as restrições de tempo devem ser passadas para o mecanismo de síntese por meio de um ou mais arquivos XDC.

Para começar, devemos definir um relógio:

- De onde vem o relógio?
 - (ou seja, porta de entrada, saída de PLL, etc.)
- Qual é o período do relógio?
 - => frequência de operação
- Qual é o ciclo de trabalho do relógio?



Definições do relógio XDC

Sintaxe do comando SDC para criar um relógio:

```
create_clock [-add] [-name <clock_name>] -period <valor> [-waveform  
<edge_list>] [<targets>]
```

O relógio, chamado meu_relógio (usado para se referir ao relógio em outros comandos), entrando no dispositivo através da portack, com período de 20 ns e ciclo de trabalho de 50% (valor padrão):

```
create_clock -período 20 -nome meu_relógio [get_ports clk] [string get_ports]-  
retorna todas as portas que correspondem à string
```

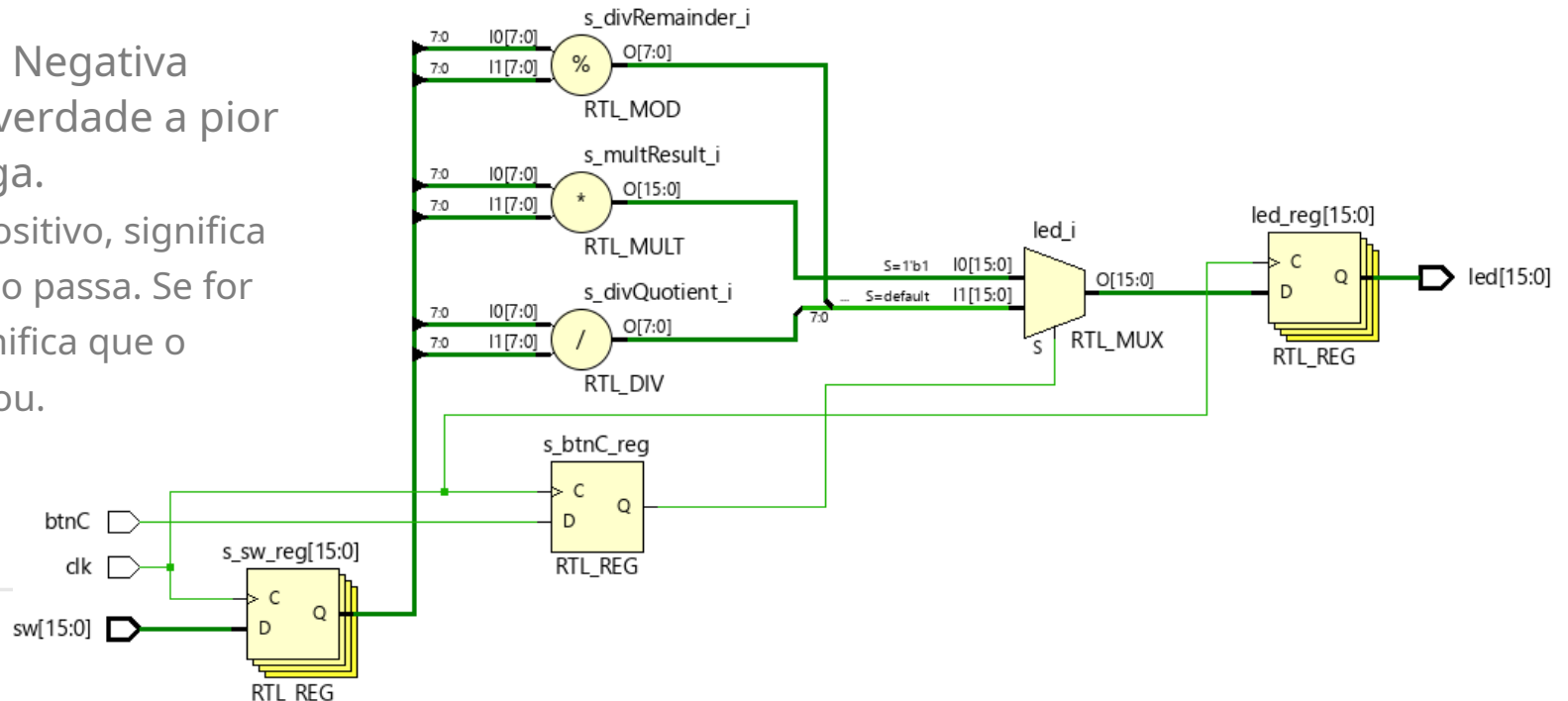
Nexys4_Master.xdc:

```
create_clock -add -name sys_clk_pin -period 10.00 -forma de onda  
{0 5} [get_ports clk]
```

Projeto Exemplo (Vivado)

A Pior Folga Negativa (WNS) é na verdade a pior (menor) folga.

Se WNS for positivo, significa que o caminho passa. Se for negativo, significa que o caminho falhou.



Report Timing Summary

- Report Clock Networks
- Report Clock Interaction
- Report Methodology
- Report DRC
- Report Noise
- Report Utilization

Report Power

Schematic

PROGRAM AND DEBUG

- Generate Bitstream
- Open Hardware Manager

Tcl Console	Messages	Log	Reports	Design Runs	Methodology	Power	Timing
Design Timing Summary							
General Information							
Timer Settings							
Design Timing Summary							
Clock Summary (1)							
Check Timing (33)							
Intra-Clock Paths							
Inter-Clock Paths							
Other Path Groups							
Setup							
Hold							
Pulse Width							
Worst Negative Slack (WNS): -5,539 ns							
Worst Hold Slack (WHS): 0,351 ns							
Worst Pulse Width Slack (WPWS): 4,500 ns							
Total Negative Slack (TNS): -50,050 ns							
Total Hold Slack (THS): 0,000 ns							
Total Pulse Width Negative Slack (TPWS): 0,000 ns							
Number of Failing Endpoints: 12							
Number of Failing Endpoints: 0							
Number of Failing Endpoints: 0							
Total Number of Endpoints: 16							
Total Number of Endpoints: 16							
Total Number of Endpoints: 34							
Timing constraints are not met.							

Frequência Máxima

A frequência máxima ($f_{\text{máx.}}$) não é fornecido explicitamente no relatório report_timing/report_timing_summary.

A frequência máxima que um projeto pode executar em hardware em uma determinada implementação:

- $f_{\text{máx.}} = 1/(t - \text{WNS})$, onde t é o período de relógio alvo.

A frequência máxima que um projeto pode executar em uma determinada arquitetura:

- O usuário terá que diminuir t e executar novamente a síntese/implementação até $\text{WNS} < 0$.
- Diferentes estratégias podem ser necessárias para obter o melhor $f_{\text{máx.}}$.

Considerações finais

Ao final desta palestra você deverá

- entenda como controlar os displays de 7 segmentos do Nexys-4
- ser capaz de seguir as seguintes recomendações:
 - Use um único sinal de clock para todos os componentes do projeto
 - Registre todos os sinais de entrada assíncronos
 - Definir restrições de tempo
 - Detectar e corrigir violações de tempo

Pendência:

- Teste os projetos fornecidos no kit Nexys-4