

MIC05: Teste de Circuitos Integrados

Marcelo Lubaszewski

PPGMicro - UFRGS

2007/II

Auto Teste Integrado

- Teste off-line
- Facilidade para teste e diagnóstico
- Geração de vetores
 - LFSR
- Análise de assinatura
 - Integração digital
 - Divisão polinomial

Tipos de Falhas

- Falhas permanentes
 - Se manifestam entre dois testes consecutivos
 - Serão detectadas apenas no próximo teste
- Falhas intermitentes
 - Acontecem sob determinadas condições que podem não ser reproduzidas durante o teste
 - Ex: para uma dada frequência de operação diferente da freq. de teste, para determinada temperatura...
- Falhas transientes
 - Acontecem de forma aleatória
 - Ex: aplicações espaciais, ruído em linhas telefônicas...

Aplicações Críticas

- Teste de manutenção é freqüente, mas:
- Se ocorre uma falha entre testes?
 - Aviação
 - Terminais bancários
 - Circuitos embarcados (automobilístico)
- No mínimo, deve-se indicar a presença de falhas, pois a resposta não é confiável

Alternativas para Aplicações Críticas

- Redundância
- Indicação das falhas é obrigatória
 - Teste on-line
 - Teste concorrente
 - Circuitos self-checking
- Em alguns casos, é preciso evitar ou mascarar as falhas
 - Tolerância a falhas

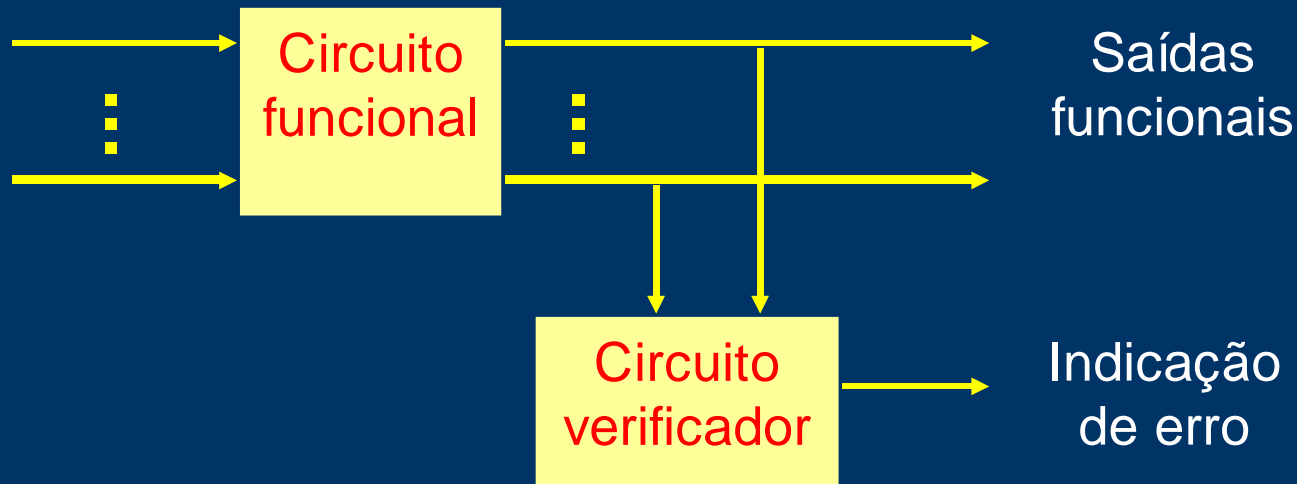
Teste On-line

Enquanto o circuito está em operação normal...

- Quem são as entradas e saídas de teste?
- Como gerar vetores?
- Como aplicar os vetores?
- Quais são as respostas esperadas?

Circuitos Self-Checking

Detecção concorrente de falhas



Teste On-line

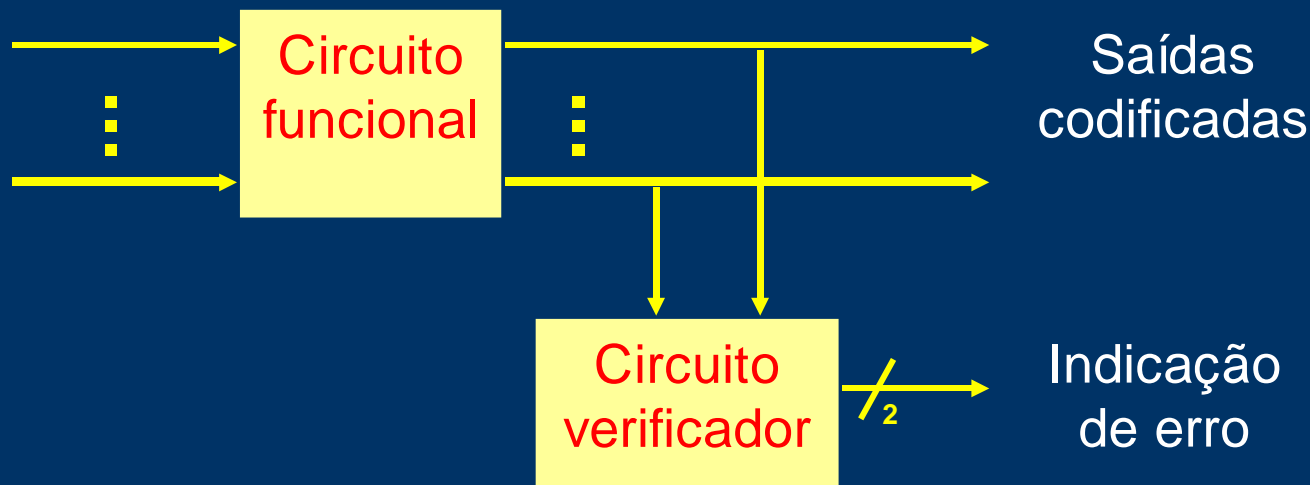
- Entradas e saídas de teste são as próprias entradas e saídas funcionais
- Vetores de teste são os próprios dados de entrada do circuito
 - Podem ou não excitar determinadas falhas
- Qual é a resposta esperada?
- Solução: codificação das saídas.
 - Uma saída fora do código previsto indica a ocorrência de falha

Circuitos Self-Checking

Detecção concorrente de falhas

Saídas codificadas e circuitos verificadores

Códigos: paridade, *hamming*, *berger*, dual



Conceitos Básicos

- Idéia básica: apenas k das 2^n combinações possíveis das saídas são válidas
- O verificador confere se a saída está entre as K válidas
- Para se ter a detecção, duas palavras pertencentes ao código devem diferir por pelo menos 2 bits (cubo)
- Distância HAMMING
 - Menor distância entre duas palavras quaisquer pertencentes ao código

Capacidade do Código

- Seja d é a distância Hamming de um código
- $d = 2e + 1$
 - Detecta $2e$ erros
 - Corrige e erros

Alguns Códigos

- Paridade
- Hamming
- Residuais
- Berger
- m-out-of-n

Paridade

- Use of parity
 - very effective single error detection
 - encoding and decoding cost is low
 - commonly used in memories, transmission over short reliable channels
 - EVEN OR ODD
 - limitations
 - unable to detect common multiple errors
 - can not be used in data transformation - for example addition does not preserve parity
- $d = ?$
- Uso em memórias

Código de Hamming – by example

- A linear block code
- Consider a (7,4) Hamming code
- Let $i_1 i_2 i_3 i_4$ be information symbols
- Let $p_1 p_2 p_4$ be check symbols
- The parity equations:

$$p_1 = i_1 \oplus i_2 \oplus i_4$$

$$p_2 = i_1 \oplus i_3 \oplus i_4$$

$$p_4 = i_2 \oplus i_3 \oplus i_4$$

- Onde colocar os bits de código?

Hamming Codes – by example (contd.)

- Can write the equations as follows (easy to remember)

p_1	p_2	i_1	p_4	i_2	i_3	i_4
1	0	1	0	1	0	1
0	1	1	0	0	1	1
0	0	0	1	1	1	1
1	2	3	4	5	6	7

This encodes a 4-bit information word a to 7-bit codeword

Hamming Codes – by example (contd.)

- Properties of the code
 - If there is no error, all parity equations will be satisfied
 - Denote the outcomes of these equation checks as c_1 , c_2 , c_4
 - If there is exactly one error, the c_1 , c_2 , c_4 point to the error
 - The vector c_1 , c_2 , c_4 is called syndrome
 - The above (7,4) Hamming code is SEC code
- $d = ?$
- What if multiple errors?

Hamming Codes – by example (contd.)

- The above method of construction can be generalized to construct an (n,k) Hamming code

- Simple bound

k = number of information bits

r = number of check bits

$n = k + r$ = total number of bits

$n + 1$ = number of single or fewer errors

Each error (including no error) must have a distinct syndrome

With r check bits max possible syndrome = 2^r

Hence: $2^r \geq n + 1$

Códigos Residuais

- How can we detect faults in the ALU?
- Parity? Hamming?
- Arithmetic code
- N information bits
- An integer k
- p code bits representing the code C where $p = N \bmod k$
- makes use of the fact
$$(M+N) \bmod k = (M \bmod k + N \bmod k) \bmod k$$

C

Códigos residuais

- $d=?$
- What if k is odd? even?
- Checker implementation cost
- The higher k , the detection capabilities vary

Código K-out-of-N

- encode each word (data/control) such that the coded word is of length n and each coded word has exactly k 1's in it
 - can detect all single errors
 - can detect all unidirectional multiple errors
- Ex: demux, decoders, etc.
- Non-separable code (there are no check bits, only information ones)

Código de Berger

- n information bits are encoded into an $n+k$ bit code word.
- The k check bits are binary encoding of the number of 1's (or 0's) in the n information bits
 - can detect all single errors
 - can detect all unidirectional multiple errors if carefully designed

Outros

- Arithmetic codes

- AN code

- used for arithmetic function unit designs
 - each data word is multiplied by a constant A
 - makes use of the identity $A(N+M) = AN + AM$
 - choice of A is important

- Checksums

- data is sent/stored with a checksum and when used the checksum is regenerated and compared to the a priori known checksum
 - functions used for checksum
 - add, exclusive-OR (bit wise), end with end around carry, LFSR, ...
 - limitation
 - can only perform (normally) error detection

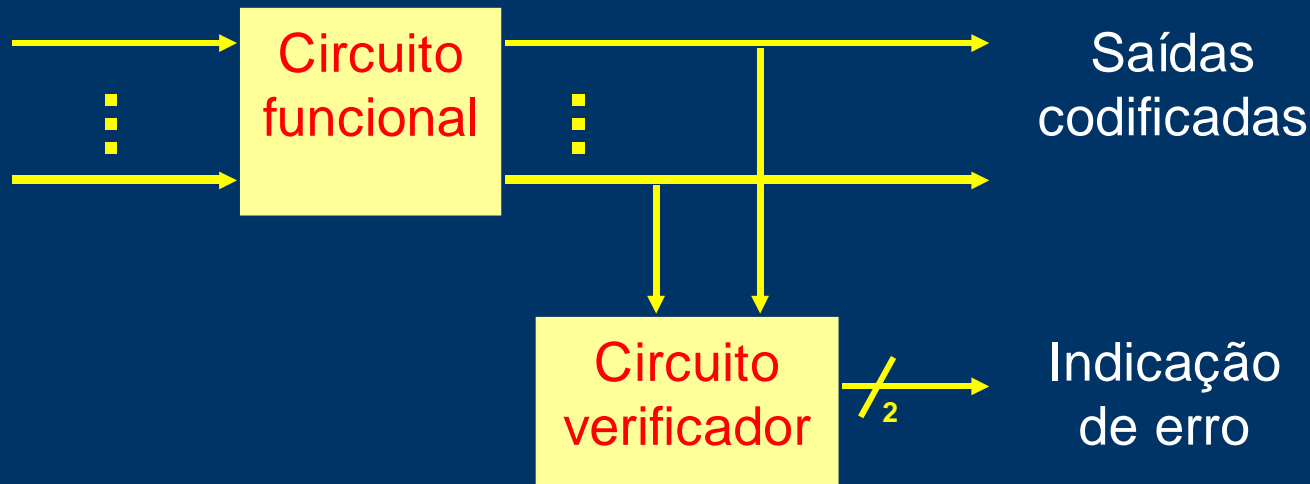
Self-checking Properties

- a circuit is *fault secure* if in the presence of a fault, the output is either always correct, or not a code word for valid input code words
 - Ensures correct behavior
- a circuit is *self-testing* if only valid inputs can be used to test it for the faults
 - Ensures fault detection during normal operation
- a circuit is *totally self-checking* if it is fault secure and self-testing

Circuitos Self-Checking

E as falhas no verificador?

Por que verificador tem duas saídas?



TSC Checkers

- Parity checker
 - Two disjoint sets of inputs generate two parity bits
- Equality checker
 - 1-out-of-2 case
- Others

Qual código?

- Definição do código:
 - Tipo e quantidade de falhas que detecta (Simulação de falhas)
 - Tipo de circuito
 - Possibilidade de correção
 - Custo de implementação do circuito verificador
 - Propriedade TSC
- Pode requerer o re-projeto

Tolerância a Falhas

- Redundância no Hardware
- Redundância de Informação
- Redundância no tempo
- Redundância por Software

Hardware redundancy

- Passive hardware redundancy
- TMR with a voter
 - main problem
 - single point of failure
 - justification
 - voter is much lower complexity and can be designed using more reliable technology
 - alternative
 - TMR with triplicated voter
- NMR voter based generalization

Hardware redundancy

(contd.)

- Active hardware redundancy
 - duplicate with comparison
 - single point of failure
 - standby sparing
 - one operational unit - it has its own fault detection mechanism
 - on occurrence of fault a second unit (spare) is used
 - cold standby - standby is in unknown state
 - hot standby - standby is same state as system - quick start
 - can generalize to n - one active and $n-1$ standby spares

Hardware redundancy (contd.)

- Hybrid hardware redundancy
 - Key - combine passive and active redundancy schemes
 - NMR with spares
 - example - 5 units
 - 3 in TMR mode
 - 2 spares
 - all 5 connected to a switch that can be reconfigured

Information redundancy

- Key concept - add redundancy to information/data
 - all schemes use Error detecting or Error correcting coding
 - Self-checking circuits

Time redundancy

- Key Concept - do a job more than once over time
 - examples
 - re-execution
 - re-transmission of information
 - different faults and capabilities of different schemes
 - transient faults
 - re-execution and re-transmission can detect such faults provided we wait for transient to subside
 - permanent faults
 - simple re-execution or re-transmission will not work.
 - faults in ALU
 - » re-execution with complement or shifted version can detect permanent and transient faults
 - » (RESO concept - re-computation with shifted operands)

Basic Fault-Tolerant Techniques

