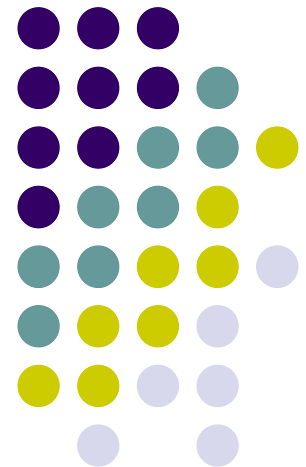
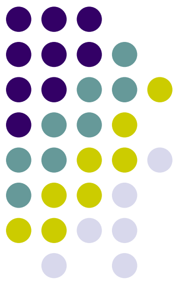
Power Conscious RTL Test Scheduling

Jaroslav Škarvada, Zdeněk Kotásek, Tomáš Herrman

skarvada@fit.vutbr.cz, kotasek@fit.vutbr.cz, herrman@fit.vutbr.cz

Brno University of Technology
Faculty of Information Technology

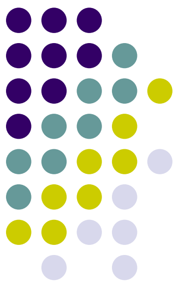




Outline

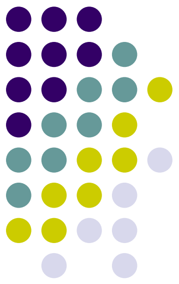
- Introduction
- Methodology Description
  - Formal Model of the Circuit
  - Testable Blocks Partitioning
  - Power Consumption Reduction
  - ILP Based Test Scheduling
- Benchmark circuits used
- Experimental results
- Conclusion

Introduction



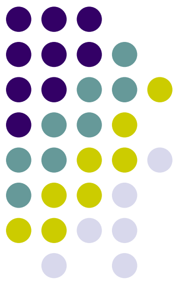
- Methodology for power conscious RTL test scheduling based on the concept of Testable blocks.
- AMI technological library is used for precise power consumption estimation.
- Test vectors and scan cells reordering approach for power consumption minimization.
- Compatibility with professional design tools.
- The methodology is targeted to pipeline circuits with low number of feedbacks.

The Outline of the Methodology



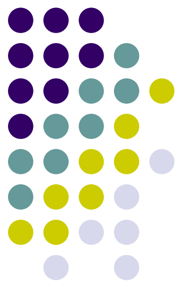
- The Circuit Under Analysis (CUA) is mapped onto AMI 1.2μm CMOS technology.
- The formal model of the CUA is created.
- The CUA is partitioned into TBs.
- Test vectors are generated for each TB.
- Power consumption is estimated for each TB
  - Number of transition counts (*NTC*) 0→1, 1→0 through test application is tracked for each node as power consumption metrics.
- Test vectors and scan cells are reordered.
  - Criteria: NTC.
- Final test is scheduled.
  - Criteria: not to overcome defined NTC limit (chip limit).

The Outline of the Methodology



- The Circuit Under Analysis (CUA) is mapped onto AMI 1.2μm CMOS technology.
- The formal model of the CUA is created.
- The CUA is partitioned into TBs.
- Test vectors are generated for each TB.
- Power consumption is estimated for each TB
  - Number of transition counts (*NTC*) 0→1, 1→0 through test application is tracked for each node as power consumption metrics.
- Test vectors and scan cells are reordered.
  - Criteria: NTC.
- Final test is scheduled.
  - Criteria: not to overcome defined NTC limit (chip limit).

Circuit under Analysis (CUA) RTL Formal Model



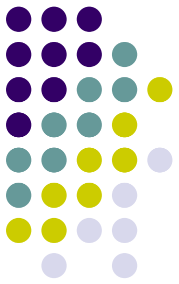
- RTL formal model reflects:
  a) CUA structural features
  b) CUA diagnostic features

In our model CUA is represented by 5-tuple:

$$CUA = (E_{CUA}, P_{CUA}, C_{CUA}, PI_{CUA}, PO_{CUA})$$

- E_{CUA} — Set of elements (Registers, multiplexors, adders, …)
- P_{CUA} — Set of ports (Inputs, outputs, no primary)
- C_{CUA} — Set of connections between ports
- PI_{CUA} — Set of primary input ports
- PO_{CUA} — Set of primary output ports

Short Introduction to the Model



- CUA elements:

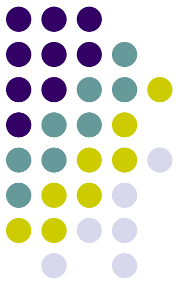
  $E_{CUA} = (R_{cua} \cup FU_{cua} \cup MX_{CUA})$

  where R_{cua} is set of all registers,
  FU_{cua} is set of all functional units (adders, subtractors, …) and
  MX_{cua} is set of all multiplexers).

- Interfaces of CUA elements:

  Let $e \in E_{CUA}, \psi(e) = \{p \mid p$ is input, output or bidirectional port of $e\}$,
  than the set of ports P_{CUA} can be defined as $P_{CUA} = \bigcup_{e \in E} \psi(e)$.
  In the model it also holds: $P_{CUA} = (IN_{CUA} \cup OUT_{CUA} \cup CI_{CUA})$,
  where IN_{CUA} is the set of all input ports (without primary) of all elements,
  OUT_{CUA} is the set of all output ports (without primary) and
  CI_{CUA} is the set of all control and synchronizing ports.

Short Introduction to the Model (cont.)

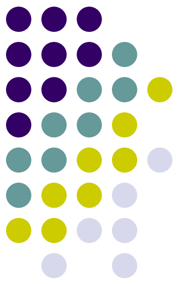


- Connections:
  - Defines the topology of the CUA.
  - The set of all connections C_{CUA} between all ports fulfils the following definition:

$$C_{CUA} \subset (PI_{CUA} \cup P_{CUA}) \times (PO_{CUA} \cup P_{CUA})$$

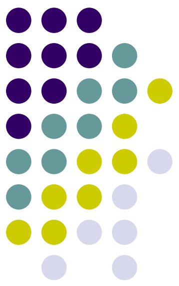
- CUA interface:
  - Defines the interface between CUA and other world.
  - The CUA interface consists of set of primary input ports PI_{CUA} and set of primary output ports PO_{CUA}.

The Outline of the Methodology



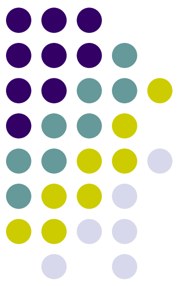
- The Circuit Under Analysis (CUA) is mapped onto AMI 1.2μm CMOS technology.
- The formal model of the CUA is created.
- The CUA is partitioned into TBs.
- Test vectors are generated for each TB.
- Power consumption is estimated for each TB
  - Number of transition counts (*NTC*) 0→1, 1→0 through test application is tracked for each node as power consumption metrics.
- Test vectors and scan cells are reordered.
  - Criteria: NTC.
- Final test is scheduled.
  - Criteria: not to overcome defined NTC limit (chip limit).

Testable Block I



- It is a segment of the CUA separated by registers from other segments and has defined interface. The separation registers are called border registers.
- The test can be applied to the TB only through its interface (border registers or *PI*/*PO*).
- The registers other than border registers must not be connected into scan chains.

Testable Block II



TB Input Interface

TB Output Interface

Border Input Registers

Testable Block

Border Output Registers

BRI_1

BRI_2

...

BRI_n

Internal logic and registers, high testability

BRO_1

BRO_2

...

BRO_n

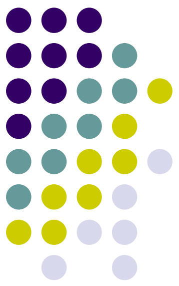
PI_1 ... PI_n

Primary Inputs

PO_1 ... PO_n

Primary Outputs

Testable Block III

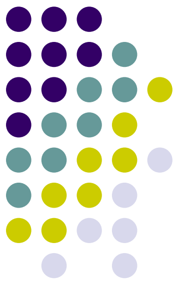


- **Set of elements in TB:**
  $E_{TB} \subseteq E_{CUA}$
- **Set of input ports of elements in TB:**
  $IN_{TB}=\{p \mid p \in P_{TB} \wedge p \in IN_{CUA}\}$,
- **Set of output ports of elements in TB:**
  $OUT_{TB}=\{p \mid p \in P_{TB} \wedge p \in OUT_{CUA}\}$,
- **Set of control and synchronizing ports of elements in TB:**
  $CI_{TB}=\{p \mid p \in P_{TB} \wedge p \in CI_{CUA}\}$
- **Set of border registers:**
  $BRO_{TB}=\{r \mid r \in R_{TB} \wedge \forall p_1, p_2: p_1 \in \psi(r) \wedge p_1 \in OUT_{TB} \wedge p_2 \in (P_{CUA} \setminus P_{TB}) \wedge p_2 \in IN_{CUA} \wedge (p_1, p_2) \in C_{CUA}\}$
  $BRI_{TB}=\{r \mid r \in R_{TB} \wedge \forall p_1, p_2 : p_1 \in (P_{CUA} \setminus P_{TB}) \wedge p_1 \in OUT_{CUA} \wedge p_2 \in \psi(r) \wedge (p_1, p_2) \in C_{CUA}\}$
  $BR_{TB}= BRO_{TB} \cup BRI_{TB}$

Set of ports in TB:

$$P_{TB} = \bigcup_{e \in E_{TB}} \psi(e) \subseteq P_{CUA}$$

Testable Block IV



Let $E_{TB} \subseteq E_{CUA}$ be the set of elements,

$P_{TB} = \bigcup_{e \in E_{TB}} \psi(e) \subseteq P_{CUA}$ be the set of ports of elements,

$$PO_{TB} = \{ po \mid po \in PO_{CUA} \wedge \exists p : p \in P_{TB} \wedge (p, po) \in C_{CUA} \} \subseteq PO_{CUA}$$

be the set of outputs which are also CUA primary outputs (can be empty),

$$PI_{TB} = \{ pi \mid pi \in PI_{CUA} \wedge \exists p : p \in P_{TB} \wedge (pi, p) \in C_{CUA} \} \subseteq PI_{CUA}$$

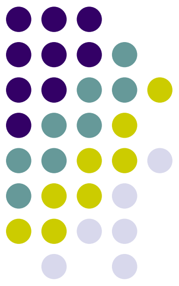
be the set of inputs which are also CUA primary inputs (can be empty),

$$C_{TB} \subset ((PI_{TB} \cup P_{TB}) \times P_{CUA}) \cup (P_{CUA} \times (PO_{TB} \cup P_{TB}))$$

be the set of internal connections and $|BRI_{TB}| + |PI_{TB}| > 0$ and $|BRO_{TB}| + |PO_{TB}| > 0$, and all the elements between the borders registers are testable (definition not given) , then the TB is determined by 5-tuple:

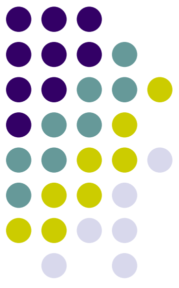
$$TB = (E_{TB}, P_{TB}, C_{TB}, PI_{TB}, PO_{TB}).$$

TB Identification Methodology

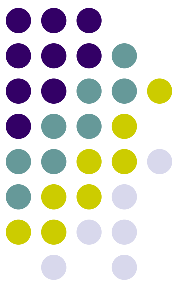


- For TB identification the genetic algorithm is used.
- All selected TBs are checked to fulfill the definition.
- Various criteria can be included in the fitness calculation.
- Border registers are candidates for scan.

Genetic Algorithm Implementation



- The binary chromosome is used.
- The length of the chromosome is equal to the number of register in CUA.
- The value of bit:
  - 0 – the register **is not** supposed to be used as a border register
  - 1 – the register **is** supposed to be used as a border register
- For each chromosome configuration it is checked how TB definitions are satisfied.
- Crossover and mutation operators are used.
- For the selection of parents roulette wheel is used.



TBs Identification Example

TB_3 TB_1 TB_2

clk i1 sel1 i2 i3 o1

1 2 3 4 5

REGISTER_D MULTIPLEXOR2 ADDITIONER2

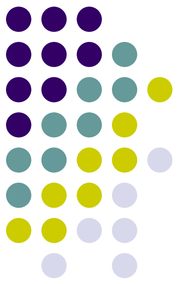
Example of chromosome: 11111

Internal representation in the software:

R_TB={ [R_1,(0,3)], [R_2,(3,1)], [R_3,(0,1)], [R_4,(1,2)], [R_5,(0,2)]}

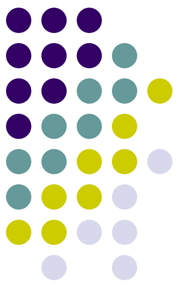
E_TB={ (MX_1,1), (ADD_1,1)}

The Outline of the Methodology



- The Circuit Under Analysis (CUA) is mapped onto AMI 1.2μm CMOS technology.
- The formal model of the CUA is created.
- The CUA is partitioned into TBs.
- Test vectors are generated for each TB.
- Power consumption is estimated for each TB
  - Number of transition counts (*NTC*) 0$\rightarrow$1, 1$\rightarrow$0 through test application is tracked for each node as power consumption metrics.
- Test vectors and scan cells are reordered.
  - Criteria: NTC.
- Final test is scheduled.
  - Criteria: not to overcome defined NTC limit (chip limit).

The Test Vectors

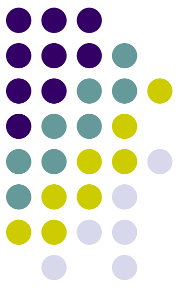


- For the test vectors generation third party professional tool is used.

The BEAST project:

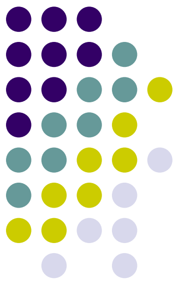
- Is used for the NTC estimation.
- Is used for the test vectors and scan cells reordering.
- Has been created in our department.
- Will be released under GPL.

Solution Space Reduction



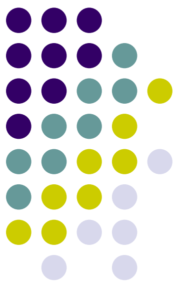
- The reordering problem is known to be NP hard with complexity of $n!$, where n is number of elements to be reordered.
- If the CUT with n_{CUT} test vectors can be partitioned into TB_1, TB_2, ..., TB_n with n_1, n_2, ..., n_m test vectors respectively and it holds that $\forall i: (i \geq 1 \wedge i \leq m \wedge m > 1) \Rightarrow (n_i < n_{CUT})$, than the problem with solution space of $n_{CUT}!$ states can be transformed to problem with solution space of $n_1! + n_2! + \ldots + n_m!$ states (that is mostly significantly smaller problem).

The Outline of the Methodology



- The Circuit Under Analysis (CUA) is mapped onto AMI 1.2μm CMOS technology.
- The formal model of the CUA is created.
- The CUA is partitioned into TBs.
- Test vectors are generated for each TB.
- Power consumption is estimated for each TB
  - Number of transition counts (*NTC*) 0→1, 1→0 through test application is tracked for each node as power consumption metrics.
- Test vectors and scan cells are reordered.
  - Criteria: NTC.
- Final test is scheduled.
  - Criteria: not to overcome defined NTC limit (chip limit).

ILP Based Test Scheduling I



- Linear programs can be expressed in canonical matrix form:

  Maximize (or minimize) $c^T x$

  Subject to $Ax \leq b$

  Where x > 0

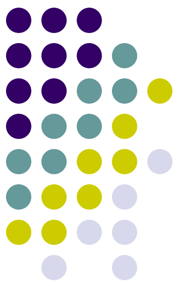
- The model of the problem:

  Let $T = \{t_1, t_2, ..., t_m\}$ represents tests for $TB_1, TB_2, ..., TB_m$.

  Let $L = \{l_1, l_2, ..., l_m\}$ denotes the length of each test (in test cycles) and gl be the bijection $T \rightarrow L$.

  Let $P = \{p_1, p_2, ..., p_m\}$ denotes the average NTC per test cycle for each test and gp be the bijection $T \rightarrow P$.

ILP Based Test Scheduling II



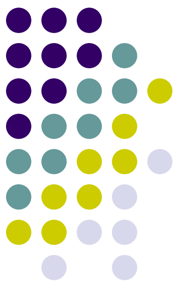
- The model of the problem:

  The goal is to partition the test to test sessions
  $S = \{s_1, s_2, ..., s_k\}$,
  $\forall s_i : s_i \in S \Rightarrow s_i \subseteq T \wedge (\cup s_i = T)$

  with minimal test application time and simultaneously not overcoming defined chip power dissipation limit P_{limit} during the test application.

  Let $X_{i,j}$, $i > 0 \wedge i \leq |T| \wedge j > 0 \wedge j \leq |S|$, be the matrix of 0-1 variables defined as follows: $X_{i,j} = 1$ if test i is scheduled to session j, otherwise $X_{i,j} = 0$, then the X is matrix of unknown variables for our ILP model.

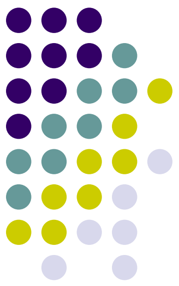
ILP Based Test Scheduling III



- ILP objective function:
  minimize: $\sum_{\forall i:s_i \in S} \max_{\forall j \in s_i} (X_{i,j}.gl(j))$
  subject to: $\max_{\forall i:s_i \in S} \left(\sum_{\forall j \in s_i} X_{i,j}.gp(j) \right) < P_{\text{limit}}$
  for $\forall i{:}s_i \in S$: $\sum_{\forall j \in s_i} X_{i,j} \leq 1$
- Non linear objective function is linearized and limiting conditions are added:
  $$\forall i \in T, \forall j \in S : C_j \geq X_{i,j}.gl(i)$$
- Final form of objective function:
  minimize: $\sum_{\forall j \in S} C_j$

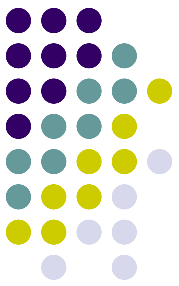
  subject to: $\max_{\forall s_i \in S} \left(\sum_{\forall j \in s_i} X_{i,j}.gp(j) \right) < P_{\text{limit}}$

ILP Based Test Scheduling IV



- The previously defined model were implemented in GNU MathProg modeling language.
- The GNU Linear Programming Kit were used for problem solving.

Benchmark Circuits

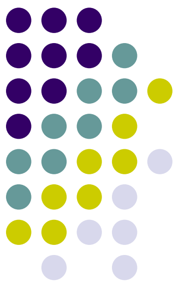


- The methodology is targeted to pipeline circuits with low number of feedbacks.
- To evaluate the methodology, the circuits from synthetically generated benchmark were used:

  http://www.fit.vutbr.cz/research/prod/index.php?id=57¬itle=1

Circuit	Number of PI / PO	Number of connections	Number of cells	Number of registers	Approx. chip area used (AMI 1.2μm) [μm^2]
k5	12 / 4	1323	58	25	147.66
k10	20 / 5	3574	113	47	280.95
k20	39 / 10	11565	226	94	561.90

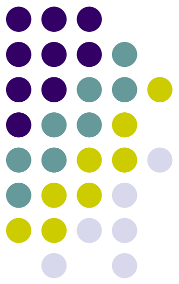
Partitioning Results



Partitioning of k5 circuit into TBs

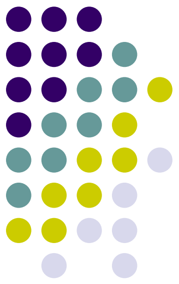
TB	PI/PO	Num of cells	Test vecs	NTC/vec	Opt. NTC/vec
k5_1	5/3	17	6	15	8
k5_2	5/3	17	6	15	8
k5_3	5/2	14	8	21	10
k5_4	3/1	9	7	13	6
k5_5	3/5	17	8	15	6

Test Scheduling Results



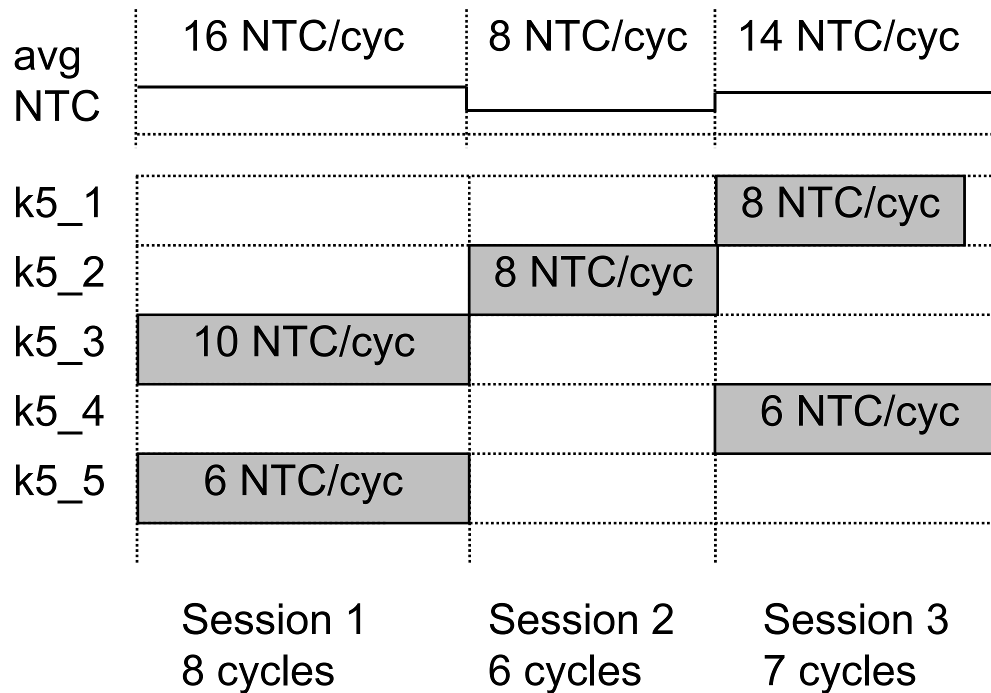
Test scheduling results for k5 circuit

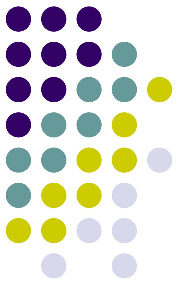
Circuit	P_{limit} [NTC/cyc]	P_{limit} [% of max]	Number of test cycles	Number of test sessions	P_{max} [NTC/cyc]
k5	19	25	–	–	–
k5opt	19	25	21	3	16
k5	39	50	21	3	36
k5opt	39	50	8	1	38
k5	59	75	14	2	49
k5opt	59	75	8	1	38
k5	80	100	8	1	79
k5opt	80	100	8	1	38



Example of the Test Schedule

Example of the test schedule for k5 circuit after test vectors reordering, P_{limit} was set to 19 NTC/cycle





Conclusions

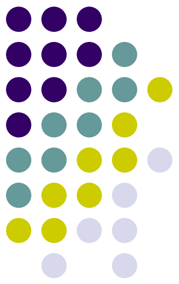
- The methodology for power constrained RTL test scheduling based on ILP was created.

Pros:

- The mathematical and formal background of the methodology.
- Precise power consumption estimation via technological library.
- Reduction of solution space for test vectors reordering.

Cons:

- The methodology is targeted to pipeline circuits with low number of feedbacks. The usage with other type of circuits is problematic (no success with ISCAS benchmarks).



Thank you for your attention.