TAM/wrapper Co-optimization And Test Scheduling For SOCs Based On Hybrid Genetic Algorithm

Xu Chuan-pei
Department of Electronic Engineering, Guilin University of Electronic Technology, Guilin, 541004, China
Email: xcp@guet.edu.cn

Lu Xue-yun and Hu Cong
Department of Electronic Engineering, Guilin University of Electronic Technology, Guilin, 541004, China
Email: lxy269054135@163.com HuCong@guet.cn

Abstract—In this paper, a new method is presented for TAM/wrapper co-optimization based on hybrid genetic algorithm and two-dimensional packing problem. In this method, core test is represented by rectangles, and a hybrid genetic algorithm that provides highly optimal solution for two-dimensional packing problem is introduced for TAM allocation and test scheduling. During the scheduling, the TAM width assigned to cores could be adjusted to an appropriate size to minimize the idle time. This HGA based method was implemented in C and applied to ITC’02 SOC Test Benchmark. Experimental results show that lower testing time was obtained by this new method compared to other methods [1,4,5].

Index Terms—SOC co-optimization, hybrid genetic algorithm, packing problem

I. INTRODUCTION

The development of SOC technology reduces the cost of electronic products. However, the highly integrated functional modules make SOC testing more complex at the same time. A lot of IC companies can not afford the expensive cost of testing. There is a contradiction between the testing time and test resources. The scarce test resources cause the difficulties of IP core parallel test. This directly lead to the consumption of very long testing time. To address this issue, researchers must develop effective test programs so as to complete the test tasks in the shortest time with using limited test resource.

Facing the challenge of the SOC test, IEEE and VSIA and other international organization formulated the SOC test criteria diligently. IEEE proposed the P1500 standard, namely embedded essence test criteria (SECT, the Standard for a Embedded Core Test). Basic idea of P1500 standard is to provide standard test structure to ensure that in SOC, IP core and the Interconnection structure between the IP cores can be tested. The test architecture of IEEE Consists of four parts: test stimuli source, test response sink, test access mechanism (TAM), core test wrapper (CTW). Test stimuli source generates test stimuli, test response sink analyzes the result of test, TAM provide a means to transmit test data. It transmit the test stimuli from the source to the wrapper of the core and transmit the test response from the wrapper of the core to the sink. Wrapper is the interface between nuclear and TAM. It not only provides the right test access path to load test stimuli and capture test response, but also ensure the isolation between tests of cores. Besides above, wrapper provides fore modes to cores: The normal function modes, the core scan test mode, reset mode of core testing, as well as the Interconnect test mode between cores. Wrapper and TAM impact directly to chip area, test time and test data volume, so it is extremely important to optimize them.

Basing on the IEEE P1500 test strategy, SOC test optimization problem can be divided into Wrapper/TAM design and test scheduling optimization, these problems have been researched by scholars, and certain solutions have been proposed. Most prior researches have studied wrapper/TAM design and test scheduling optimization as independent problems [4,6]. However, there is close relationship between Wrapper/TAM design and test scheduling optimization, solving these two problems in conjunction will certainly shorten the testing time.

In this paper, Best Fit Decreasing (BFD) algorithm is used to completed the design of wrapper to minimize the testing time of each IP core, and hybrid genetic algorithm is used to solve the problem of TAM design and test scheduling. The method which combine HR algorithm with design of flexible width TAMs is used to calculate the fitness of chromosome.

Experimental results show that lower testing time was obtained by our method compared to other methods [1,4,5].
II. RELATED CONCEPTION AND PRIOR WORK

The P1500 wrapper is a shell around a core, that allows that core to be tested as a stand-alone entity by shielding it off from its environment. Like wise, the wrapper allows the environment to be tested independent from the state of the core\(^{10}\).

The issue of designing balanced scan chains within the wrapper was addressed in \([11]\). Fig. 1 shows two connection mode between wrapper I/O unit and scan chain2, 1(a) is an unbalanced wrapper, 14 clock cycles is needed for running a test stimulus. 1 (b) is a balanced wrapper, 8 clock cycles is needed for running a test stimulus \([5]\). This shows that balancing scan chains can reduce the testing time of given core.

TAM is test access mechanisms which transports the stimuli from source to core and from core to sink. A wrapper that amongst other things, connects the TAMs to the core\(^{10}\). During the scheduling each IP core should be connected to a TAM of certain width. The major issue in test scheduling is determining the TAMs width that assigned to each IP core. Test scheduling for SOCs involving multiple test resources and cores with multiple tests is especially challenging. Even simple test scheduling problems for SOCs have been proved to be NP-hard. Many methods have been used to solve test optimization problem. In prior studies, modus operandi is dividing a test bus of certain width into several groups, the wrapper of IP core was connected to a certain group of bus, each IP core use a fixed TAM width throughout the testing process. This is referred as fixed-width TAM architecture. For example, bus is divided into 2 parts in \([12]\), then Integer Linear Programming (ILP) algorithm was used for test scheduling. In \([4]\), bus is divided into 2 or 3 parts and then ILP algorithm is replaced by Genetic algorithm(GA). GA gives very good results, however there is great room for improvement if flexible-width TAM architecture is adopted.

V. Iyenga introduced a flexible TAM architecture in 2002\(^{1}\). It was shown in \([1]\) that for a given core, the testing time varies with TAM width as a “staircase” function. This implies that the testing time does not decrease with increasing TAM width until a core-specific threshold is exceeded. Hence if a core is connected to a TAM of width \(w\), the same testing time may actually be obtained using only \(w'\) wires (\(w\leq w'\)). The remaining \(w-w'\) wires, which could have been used to transport test data for another core, are not efficiently utilized. So only threshold width should be choosed to assigned to cores. This is the basis on which the V. Iyenga determines the TAM width for each core.

The method in \([1]\) use BFD algorithm to design a set of wrappers for each core to eliminate the mismatch between the core’s test data. A test schedule is then determined by test scheduling algorithm and a temporary TAM width is assigned to each core, this temporary TAM width can be adjusted in accordance with specific needs during the test scheduling. In this method, there is no explicit partition of the total TAM width. Instead, a more flexible TAM architecture is created in which the total TAM width is partitioned effectively among the group of cores being tested during any time interval in the schedule. This partition is allowed to vary with time.

This flexible TAM architecture can significantly reduce the waste of test resources. However, the method in \([1]\) didn’t obtain desired result because ILP algorithm is used for test scheduling. Studies have proved that ILP algorithm is not very effective for solving test scheduling problem.

Addressing all issues above, flexible-width TAM architecture is combined with hybrid genetic algorithm to complete the co-optimization in this paper.

III. SOC TEST OPTIMIZATION MODEL

The co-optimization problem in this paper is as follows:

Given the total TAM width \(W_{\text{max}}\) for the SOC, the wrapper and TAM assignment should be determined for each core, and a test schedule should be designed for the SOC, such that:

a. The total number of TAM wires utilized at any moment does not exceed \(W_{\text{max}}\).
b. The overall SOC test completion time is minimized.
In this paper, co-optimization is transformed into rectangle packing problem. The rectangle packing problem is described as follows:

Given a collection of rectangles, and a bin of fixed width and unbounded height, pack the rectangles into the bin, such that no two rectangles overlap, and the height of the bin is minimized.

In the co-optimization model, the test of each IP core will be represented by a rectangle. The width of the rectangle corresponds to the TAM width assigned to the core, and the height of the rectangle corresponds to the testing time of the core.

When core test is represented with rectangle, two parameters must be initialized: the width and height of each rectangle, namely the width of bus assigned to each IP core and the test time of the core for this width. The method described in [1] is used to initialize these two parameters. The pseudocode for the initialization is presented in Fig. 4 and the definition of highest Pareto-optimal width can be seen in [1].

The parameters in the pseudocode are as follows:
- \( w_{pi} \)---preferred TAM width for core \( i \);
- \( T_{pi} \)---testing time of core \( i \) for the width of \( w_{pi} \);
- \( T_{i}(w) \)---testing time of core \( i \) for the width of \( w \);
- \( w_{i} \)---the width of rectangle for core \( i \);
- \( h_{i} \)---height of rectangle for core \( i \);
- \( W_{max} \)---the TAM total width;
- \( p_{c} \)---control parameter, control the value of \( T_{i} \) and thus control the value of \( w_{i} \);
- \( d_{c} \)---control parameter, control the equivalence probability between \( w_{pi} \) and highest Pareto-optimal width;

In this paper, BFD algorithm is used to design a set of wrappers for each core. A preferred TAM width is assigned to each core in line 1-9, and then BFD algorithm is used for calculating the test time of each core for the width of \( w_{pi} \). Width and height of rectangle \( i \) are made respectively equal to the preferred TAM width and test time of the core \( i \).

A set of rectangles for core test can now be constructed and the test scheduling is transformed into a process of packing, the goal of optimization is to pack all rectangles into a bin of fixed width (\( W_{max} \)) with using a smallest height of the bin, namely using the shortest testing time.

### IV. CO-OPTIMIZATION BASED ON HGA

In this paper, hybrid genetic algorithm [7] is used to solve the problem of test scheduling. First, the search ability of genetic algorithm is used to find the order in which IP cores are connected to the bus, namely the order in which rectangles are filled into the box. Then fitness value of chromosome is calculated by the method described in section 4.1. After several iterative operations, the best result of HGA is chosen as the solution for test scheduling problem.

Construction of hybrid genetic algorithm is divided into two parts: design of fitness function and design of genetic operators.

#### A. Fitness Function

In this paper, the value of fitness function of chromosome is the height of the box that has been filled, namely the testing time of SOC. The flexible-width TAM architecture is combined with HR algorithm to calculate the value of fitness function.

The rule of HR algorithm is as follows [6]:
1. Create a bounded space. First, rectangles are sorted in the order that generated by GA. Then the rectangle which ranks first is filled into the box. At this point a bounded space \( S \) is created and this space is seen as the current sub-space waiting for filling.
2. Pack a rectangle into the bounded space, then the space is divided into two subspaces Fig. 6).

```plaintext
1. compute collection \( A \) of cores of SOC
2. Set \( g=\) the quantity of cores belong to \( C \)
3. FOR \( i=0 \) to \( g \)
4. Set \( T_{i}(W_{max})=BFD(i, W_{max}), T_{i}(1)=BFD(i, 1) \)
5. calculate \( T_{i}=T_{i}(W_{max})+(p_{c}/100)\times(T_{i}(1)-T_{i}(W_{max})) \);
6. Set \( w_{pi}=w \), such that \( T_{i}(w)-T_{i} \) is minimum
7. Calculate highest Pareto-optimal width \( w_{h} \):
8. IF \( w_{pi}=w_{h} \leq d_{c} \)
9. Set \( w_{pi}=w_{h}, T_{pi}=BFD(i, w_{pi}) \),
10. END
11. END
12. FOR \( i=0 \) to \( g \)
13. Set \( w_{i}=w_{pi}, h_{i}=T_{pi} \)
14. END
```

Figure 4. Initialization for rectangle

© 2010 ACADEMY PUBLISHER
(3). Pack each subspace by the method in (2) recursively.
(4). Create a new space for the remaining rectangles (Fig. 7).

If all the sub-space has been filled while there are still rectangles remaining, a new bounded space will be created with the method proposed in (1) for the remaining rectangles, then the remaining rectangles will be packed into the new space by the method proposed in (2) and (3).

Some spaces remain blank because they are too small to be filled with any rectangle during packing. That results in a waste of space. The following method is used to reduce waste:

(1). Insert rectangle to fill idle\(^1\). Width and height of rectangles can be adjusted because flexible-width TAM architecture is adopted. At this time a rectangle is selected to be inserted into the gap after being assigned a new width, then the waste of space can be reduced. In practice, the width of the gap corresponds to the width of the idle bus, and the height of the gap corresponds to the idle time of the bus. The preferred TAM width that selected core obtained before is replaced by the width of idle bus, then this IP core can be connected to the idle bus. After being connected to the bus, the testing time of the selected core will be calculated by BFD algorithm.

Let Q be the set of IP cores meeting the condition dif\(_j\); > 0, the selected core should meet the following condition:

\[
\text{dif}_{\text{selected}} < \text{dif}_{\text{j}} (\text{selected} \in Q, j \in Q, \text{selected} \neq j) \quad (1)
\]

Where:
- \(idle_t\)---idle time of bus;
- \(new_t_i\)--- the testing time of core \(i\) after core \(i\) being connected to bus;
- \(\text{dif}_i = \text{dif}_i = idle_t - new_t_i\); (2). Widen rectangle to fill idle\(^1\). If the suitable rectangle can not be found in previous step, a filled rectangle will be selected to obtain an extra width of \(idle_w\), so that its height is decreased (figure 9). In practice, an IP core that have been connected to bus is selected to obtain an extra width of \(idle_w\), then test time of this IP core will be shortened. After obtaining an extra width, the testing time of the selected core will be calculated by BFD algorithm.

Let P be the set of IP cores meeting the condition that \(\text{begin}_i = idle_{begin}\), the selected core should meet the following condition:

\[
D_{\text{sel}} > D_i (\text{sel} \in P, j \in P, \text{selected} \neq j) \quad (2)
\]

Where:
- \(idle_{begin}\)---time when bus enters idle status;
- \(\text{begin}_i\)---time when the test of core \(i\) begin;
- \(\text{orig}_t_i\)--- testing time of core \(i\) before it obtaining the extra width;
- \(\text{new}_t_i\)--- testing time of core \(i\) after it obtaining the extra width;
- \(\text{D}_{\text{sel}} = \text{orig}_t_i - \text{new}_t_i\); (2). Widen rectangle to fill idle\(^1\). If the suitable rectangle can not be found in previous step, a filled rectangle will be selected to obtain an extra width of \(idle_w\), so that its height is decreased (figure 9). In practice, an IP core that have been connected to bus is selected to obtain an extra width of \(idle_w\), then test time of this IP core will be shortened. After obtaining an extra width, the testing time of the selected core will be calculated by BFD algorithm.

Combined With the two methods used to reduce waste, fitness function calculation process is shown in Fig 10. At the end of packing, fitness value will be returned to the genetic algorithm. The SOC testing time is equal to the height of the box that has been filled.
B. Realization Of HGA

Genetic algorithm is a simulation of genetic selection and natural selection in biological evolution process. It is a algorithm with the function of “survival + detection”. All the individuals in a group can be seen as the object of a algorithm with the function of “survival + detection”, and natural selection in biological evolution process. It is solution for Large-scale problems. Studies have shown that Hybrid genetic algorithm, which of local search, and it prone to premature convergence. However, GA has it’s robustness and search capability. Genetic algorithm has been widely applied in various fields because of it’s strong kernel of genetic algorithm. Genetic algorithm has been done by genetic operations, settings the control parameter are the creation, the design of fitness function, the design of operation and parameter encoding, initial population, Selection, crossover, mutation constitute the genetic GA, and an encoded parameter space efficiently searched.

### Step 1: Population initialization
Create a sequence coded chromosome with the length of POPSIZE. The chromosome is used as the first chromosome. Two random breakpoints are selected and the genes between these two breakpoints are swop. The value of the new chromosome is lower than the old chromosome, the new chromosome is accepted as the offspring; otherwise the old chromosome is accepted as the offspring.

### Step 2: Calculate the fitness value of individuals of population.
Fitness function can be calculated by the method described in 4.1. Detailed process of HGA is as follows:

#### Start
Create a bounded space

#### Pack rectangle

#### Insert rectangle

#### Widen rectangle

#### End of sub-space filling ?

#### Y

#### N

#### End of packing ?

#### Y

### Step 3: Crossover
Two individuals are randomly selected from the old generation. Two new chromosomes are generated from the couple of selected chromosomes by using the method of single-point crossover. This operation was executed POPSIZE / 2 times to get the new generation.

### Step 4: Mutation
For each chromosome, two random genes on both sides of the breakpoints are exchanged. If the fitness value of the new chromosome is lower than the old chromosome, the new chromosome is accepted as the offspring; otherwise the old chromosome is accepted as the offspring.

### Step 5: Optimal preservation strategy
Chromosome with the highest fitness value in new generation is replaced by the chromosome with the lowest fitness value in old generation.

### Step 6: If the number of iterations reaches MAXG, return to step2; otherwise, the shortest time is return and the corresponding chromosome is selected as the solution of co-optimization.

MAXGENS is the largest number of iterations, POPSIZE is the population size, PXOVER is the crossover probability. In this paper, MAXG=100-400, POPSIZE=50, PXOVER=0.8.

V. EXPERIMENTAL RESULTS

Our algorithms is implemented in C, and experiments is conducted on Intel PentiumIV, 2.5G Hz processor with 256 MB memory.

In this section, experimental results are presented. These results are for three benchmark SOCs: d695, p93791, p22810. d695 is an academic benchmark which contains 3 memory cores and 8 scan-testable logic cores. p93791 is the largest example SOC which contains 18 memory cores and 14 scan-testable logic cores. SOC p22810 contains 6 memory cores and 22 scan-testable logic cores. All possible integer values of the parameters pc and dc in the range 0<pc<10, 0<dc<4 are considered and the best results are shown in tables in appendixes.

In the tables, the total TAM width is proposed in the first column, the SOC test time with using our method is proposed in the second column, the SOC test time with using the method in [1,4,5] is proposed respectively in the 3th,5th,7th column. The percentage difference between testing time obtained by our method and other methods is calculated by the formula \( \Delta T = (T_a - T_{new}) / T_a \), where \( T_a \) is the SOC testing time with using other method and the Tnew represents the SOC testing time obtained by our method.

The best results for d695 and 93791 were obtained by GA and Quantum-inspired evolutionary algorithm (QEA) when the bus is divided into three part, these results are made to compare with our method. For p22810, related literature gives the results only with the bus width partition two, so the comparison between our results and the GA results with the bus width partitions two is made. The best QEA results is obtained when the bus is divided into four parts, these results are made to compare with our method.

It can be seen from the tables that compared with the method which use the flexible-width TAM architecture, the new method proposed in this paper obtain lower testing time in almost all instance. This shows Advantage of hybrid genetic algorithm compared with ILP algorithm. Compared with the GA algorithm and the QEA algorithm, our method obtains lower testing time in most
instances for d695 and p22810. However, for p93791, our method obtain comparable or higher testing time. A careful Analysis reveals reasons for this poor result:

(1). It is difficult to optimize the system testing time with using only a single value of pc and dc for all cores.

(2). The scale of the problem is so large that the algorithms need long computation time, so the iteration number is set to 100, this results that genetic manipulation has been terminated before the best solution obtained. To solve the problem related to these reasons is the direction for further work. The control parameters pc and dc should be distributed for each core According to specific circumstances of the core. The computing time of hybrid genetic algorithm should be reduced and iteration number should be increased. And different selection probability a mutation probability should be tried.

The pie chart in Fig. shows the result after statistics all experimental data. Each part of the pie represents the proportion that one method obtains the best result among 4 methods. For example, In the 21 experiments, Our method 7 times obtains the best results compare with 4 methods. For example, In the 21 experiments, Our method in most cases, followed by our method. After tried.

It can be seen from the pie chart that QEA obtains best result in most cases, followed by our method. After the observation, the reasons why QEA performances so great advantages is found: our method obtain higher testing time for p93791. Reasons for the poor results and improved methods for this flaw have been described above and our methods will be improved to achieve better results in future work.

VI. CONCLUSION

A new co-optimization method for SOC test is proposed in this paper. In the new method Wrapper/TAM optimization and test scheduling problem are transformed into rectangle packing problem. And then hybrid genetic algorithm is used to search the best option. At last our co-optimization technique is applied to an academic benchmark SOC and two industrial SOCs. Compared with other method, lower test time can be obtained by our method in. A careful analysis reveals the reasons for the poor results and to solve the problem related to these reasons is the direction for our further work.

REFERENCES


APPENDIX EXPERIMENTAL RESULTS FOR 3 SOCS

<table>
<thead>
<tr>
<th>W max</th>
<th>Tnew</th>
<th>T[1]</th>
<th>△T</th>
<th>GA</th>
<th>△T</th>
<th>QEA</th>
<th>△T</th>
</tr>
</thead>
<tbody>
<tr>
<td>16</td>
<td>42703</td>
<td>43723</td>
<td>-2.332%</td>
<td>42268</td>
<td>+1.029%</td>
<td>42268</td>
<td>+1.029%</td>
</tr>
<tr>
<td>24</td>
<td>29432</td>
<td>30317</td>
<td>-2.919%</td>
<td>28389</td>
<td>+3.674%</td>
<td>28292</td>
<td>+4.029%</td>
</tr>
<tr>
<td>32</td>
<td>21523</td>
<td>23021</td>
<td>-6.507%</td>
<td>21518</td>
<td>+0.023%</td>
<td>21518</td>
<td>+0.023%</td>
</tr>
<tr>
<td>40</td>
<td>18380</td>
<td>18459</td>
<td>-0.428%</td>
<td>17677</td>
<td>+3.977%</td>
<td>17677</td>
<td>+3.977%</td>
</tr>
<tr>
<td>48</td>
<td>15493</td>
<td>15698</td>
<td>-1.305%</td>
<td>16975</td>
<td>-8.730%</td>
<td>16975</td>
<td>-8.730%</td>
</tr>
<tr>
<td>56</td>
<td>13407</td>
<td>13415</td>
<td>-0.059%</td>
<td>13795</td>
<td>-2.812%</td>
<td>13207</td>
<td>+1.514%</td>
</tr>
<tr>
<td>64</td>
<td>11978</td>
<td>11604</td>
<td>+3.223%</td>
<td>12941</td>
<td>-10.332%</td>
<td>12941</td>
<td>-10.332%</td>
</tr>
</tbody>
</table>

Xu Chuanpei: Ms., born in 1968, a doctorate earned from Xi’an University of Electronic Science in 2006.
She is a professor of Guilin University of Electronic Science and Technology. Her research interests are in the area of IC testing theory and technology. She has published over 30 articles in some important magazines, e.g., Acta Electronica Sinica, Chinese Journal of Scientific Instrument, Journal of Electronics & Information Technology.

Prof. Xu is a member of IEEE.

Lu Xueyun: Ms., born in 1984, a bachelor's degree earned from Guilin University of Electronic Science and Technology in 2007.
She is a graduate student of Guilin University of Electronic Science and Technology. Her research interests are in the area of integrated circuit test.

HuCong: Mr., born in 1981, master's degree earned from Guilin University of Electronic Science and Technology in 2006.
He is an instructor of Guilin University of Electronic Science and Technology. His research interests are in the area of Software Testing and validation. He has published several articles in some important magazines, e.g., Microcomputer Information, and Electronic Instrumentation Customer.
### TABLE 2. Experimental results for p93791

<table>
<thead>
<tr>
<th>W_{max}</th>
<th>Tnew</th>
<th>T[1]</th>
<th>ΔT</th>
<th>GA</th>
<th>ΔT</th>
<th>QEA</th>
<th>ΔT</th>
</tr>
</thead>
<tbody>
<tr>
<td>16</td>
<td>1842658</td>
<td>1851135</td>
<td>-0.458%</td>
<td>1751423</td>
<td>+5.209%</td>
<td>1743162</td>
<td>+5.708%</td>
</tr>
<tr>
<td>24</td>
<td>1244061</td>
<td>1248795</td>
<td>-0.379%</td>
<td>1180948</td>
<td>+5.344%</td>
<td>1173202</td>
<td>+6.040%</td>
</tr>
<tr>
<td>32</td>
<td>948601</td>
<td>975016</td>
<td>-2.709%</td>
<td>893313</td>
<td>+6.189%</td>
<td>877066</td>
<td>+8.156%</td>
</tr>
<tr>
<td>40</td>
<td>754843</td>
<td>794020</td>
<td>-4.934%</td>
<td>723400</td>
<td>+4.347%</td>
<td>705524</td>
<td>+6.990%</td>
</tr>
<tr>
<td>48</td>
<td>593328</td>
<td>627934</td>
<td>-5.511%</td>
<td>593841</td>
<td>-0.086%</td>
<td>590525</td>
<td>+0.475%</td>
</tr>
<tr>
<td>56</td>
<td>534997</td>
<td>568436</td>
<td>-5.883%</td>
<td>511001</td>
<td>+4.696%</td>
<td>508459</td>
<td>+5.219%</td>
</tr>
<tr>
<td>64</td>
<td>501661</td>
<td>511286</td>
<td>-1.883%</td>
<td>458148</td>
<td>+9.498%</td>
<td>444104</td>
<td>+12.960%</td>
</tr>
</tbody>
</table>

### TABLE 3. Experimental results for p22810

<table>
<thead>
<tr>
<th>W_{max}</th>
<th>Tnew</th>
<th>T[1]</th>
<th>ΔT</th>
<th>GA</th>
<th>ΔT</th>
<th>QEA</th>
<th>ΔT</th>
</tr>
</thead>
<tbody>
<tr>
<td>16</td>
<td>384082</td>
<td>452639</td>
<td>-15.146%</td>
<td>461348</td>
<td>-16.748%</td>
<td>438619</td>
<td>-12.434%</td>
</tr>
<tr>
<td>24</td>
<td>263303</td>
<td>307780</td>
<td>-14.451%</td>
<td>361326</td>
<td>-27.129%</td>
<td>298914</td>
<td>-11.913%</td>
</tr>
<tr>
<td>32</td>
<td>201320</td>
<td>246150</td>
<td>-18.212%</td>
<td>313891</td>
<td>-35.863%</td>
<td>230813</td>
<td>-12.778%</td>
</tr>
<tr>
<td>40</td>
<td>170162</td>
<td>197293</td>
<td>-13.752%</td>
<td>287260</td>
<td>-40.764</td>
<td>198184</td>
<td>-14.141%</td>
</tr>
<tr>
<td>48</td>
<td>151126</td>
<td>167256</td>
<td>-9.644%</td>
<td>268512</td>
<td>-43.717%</td>
<td>174496</td>
<td>-13.393</td>
</tr>
<tr>
<td>56</td>
<td>143325</td>
<td>145417</td>
<td>-1.439%</td>
<td>271429</td>
<td>-47.196%</td>
<td>151886</td>
<td>-5.636%</td>
</tr>
<tr>
<td>64</td>
<td>132170</td>
<td>136941</td>
<td>-3.484%</td>
<td>260645</td>
<td>-49.291%</td>
<td>137532</td>
<td>-3.899%</td>
</tr>
</tbody>
</table>