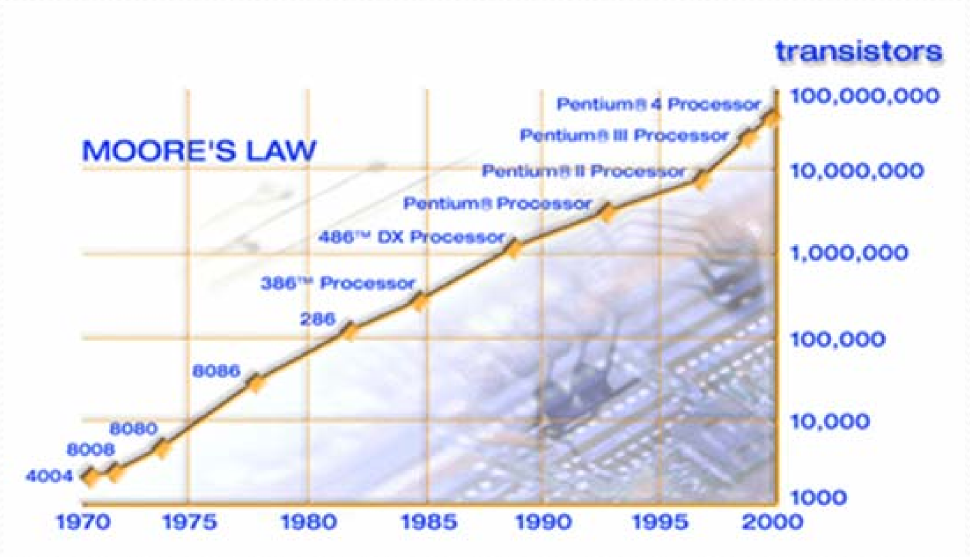
Modeling memory resources distribution on multicore processors using games on cellular automata lattices

Michail-Antisthenis I. Tsompanas, Georgios Ch. Sirakoulis* and Ioannis Karafyllidis
Department of Electrical and Computer Engineering
Democritus University of Thrace, DUTH
67100, Xanthi, Greece
{mtsompan, gsirak, ykar}@ee.duth.gr

INTRODUCTION

The problem : Moore’s Law.



A possible solution is multi-core processors…
But will this solution give an answer to the need of greater performance or it will generate more problems???

One major set of changes to platform design will be in the memory hierarchy. Research in these areas includes work on shared distributed caches, cache policies (including data-specific policies), and cache partitioning.

INTRODUCTION

- Our work focuses on the study of methods on how to distribute the memory resources between the cores of a processor.
- It is really interesting, when the cores of the processor conflict, under game theory's spectrum, for the use of the on-chip memory in order to maximize their performance.
- Consequently, game theory, which is defined as the formal study of conflict and cooperation, comes into the equation.
- On the other hand, inspired by the cores' local interaction, each core of the under study processor can be represented as a Cellular Automata (CA) cell and, more specifically, as a player in a community with a predefined number of CA neighbors, who will conflict for the occupancy and procession of local resources.

GAME THEORY AND CELLULAR AUTOMATA

- Game theory is a mathematical discipline that studies the situations where the fate of each participant depends not only on the decisions it made, but also on the decisions made by other participants.
- Regarding CA models, are very effective in simulating physical systems and solving scientific problems, because they can capture the essential features of systems where global behavior arises from the collective effect of simple components which interact locally .
- In general, a CA requires:
  - (i) a regular lattice of cells covering a portion of a d–dimensional space;
  - (ii) a set $C(\vec{r},t)=\{C_1(\vec{r},t), C_2(\vec{r},t), \ldots, C_m(\vec{r},t)\}$ of variables attached to each site of the lattice giving the local state of each cell at the time t=0, 1, 2, …; and
  - (iii) a rule R={R_1, R_2,…,R_m} which specifies the time evolution of the states $C(\vec{r},t)$ in the following way: $C_j(\vec{r},t+1)=R_j(C(\vec{r},t), C(\vec{r}+\vec{\delta}_1,t), C(\vec{r}+\vec{\delta}_2,t), \ldots, C(\vec{r}+\vec{\delta}_q,t))$ where $\vec{r}+\vec{\delta}_k$ designate the cells belonging to a given neighborhood of cell $\vec{r}$.

GAME THEORY AND CELLULAR AUTOMATA

- As proposed possible CA rule for the application of game theory to CA, the Prisoner's Dilemma, results in fine candidate.
- In order to play a single round of the Prisoner's Dilemma, the two players A and B, have but two options and must decide whether they will cooperate or defect.

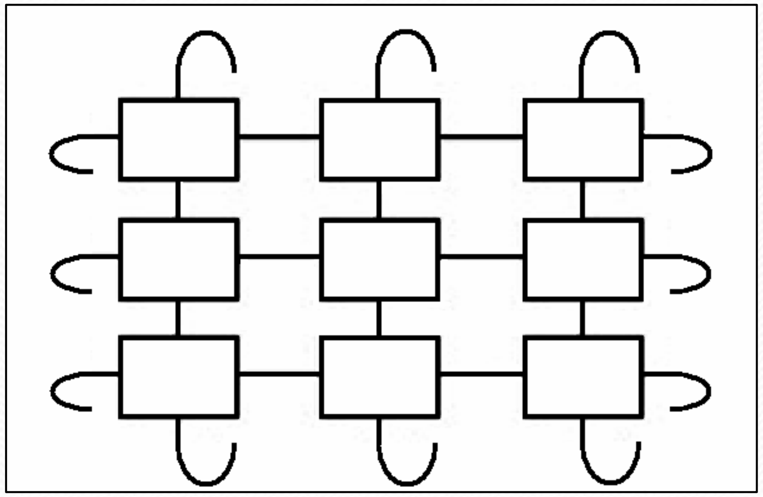
		B	
		Cooperate (C)	Defect (D)
A	Cooperate (C)	3/3	0/5
	Defect (D)	5/0	1/1

$$T > R > P > S \qquad T + S < 2R$$

- Each processor core is issued as a player that wants to take under control some of the common local resources and more specifically memory.
- There are two possible moves, as above, defect and cooperate. When one defects, it needs more resources than the predefined in its possession. When one cooperates, does not need the resources predefined to its account and can give them away.

SIMULATION

- A simulation environment has been developed using MATLAB, which generates the results from a Spatial Iterated Prisoner's Dilemma game on CA lattice.
- The player CA cells, representing the cores of a processor, are placed on a square grid, in order to have four neighbors each.
    - All cores are considered to be identical in a homogeneous multi-core system.
    - Moreover, player cells placed on the borders of the grid use as neighbors the ones placed on the opposite border, meaning periodic CA boundary conditions.



SIMULATION

- During a time step, i.e. round, every CA cell interacts with all its neighbors and at the end of that round collects the payoffs it gained and sums those to its total score achieved from the earlier rounds.
- If S(n) is the total score a player has achieved until round n and P_1, P_2, P_3 and P_4 the payoffs from the interaction between the player and each one of its neighbors on that round, then its total score on the round n+1 will be:

$$S(n+1)=S(n)+P_1+P_2+P_3+P_4$$

- Each core of the processor, i.e. CA cell, can potentially have its own strategy that dictates it what kind of move it will choose on every round of the game.

SIMULATION

The strategies each player can follow are the five most debated amongst game theorists.

- Defective strategy is the one that the player always chooses to defect, which represents a core that needs to use more resources.
- Cooperative strategy is the one that the player always chooses to cooperate, which represents a core that does not need any more resources.
- Random strategy is the one that the player chooses randomly to defect or to cooperate, simulating a core in a real-time situation in which sometimes needs resources and sometimes does not.
- Tit-for-Tat strategy, which is the one that a player cooperates on the first move and then does exactly what the other player did on the previous move, and
- Pavlov strategy, which is the one that the player repeats its former choice whenever it earns a high payoff like 5 or 3 and switches that choice whenever it earns a low payoff like 1 or 0.

These two last "rational" strategies are used to illustrate the possibility of an alternation of the results by using logical players (using memories to make their next move).

SIMULATION

Moreover seven possible strategy swap scenarios have been taken under consideration corresponding to possible cores' attribution depicted in the CA grid.

(i) No swapping, when all players maintain the strategy assigned to them from the beginning.

(ii) Synchronous updating (SU), when at the end of each round, the scores of all the neighbors of each player are evaluated and the strategy of the one with the highest score is adopted. The changes of all the strategies happen in parallel.

(iii) Synchronous updating after five rounds (SUFR), which is the same scenario as above, but it occurs after five rounds of initiation.

(iv) Random asynchronous updating with replacement (RAUWR), when at the end of each round and for N (the number of players) micro-time-steps, a player is selected at random from the community, and updated. As a consequence, as all players are updated, they "awake" to see a slightly different world from that of the cells updated before and after them.

SIMULATION

(v) Random asynchronous updating without replacement with lost steps (RAUWRWLS), at which, for each micro-time-step, a player is chosen at random and updated. Unlike the random asynchronous updating with replacement scenario, once a player is updated he cannot be updated again even if he is chosen again.

(vi) Random asynchronous updating without replacement (RAUWOR), which is identical to the random asynchronous updating without replacement scenario with lost steps; however modules can be chosen only once, thus every single player cell is updated.

(vii) The last swapping scenario is the Random asynchronous updating with a fixed order (RAUWFO), which is the same as the random asynchronous updating without replacement scenario; however, players are updated in a fixed random order throughout the entire simulation.

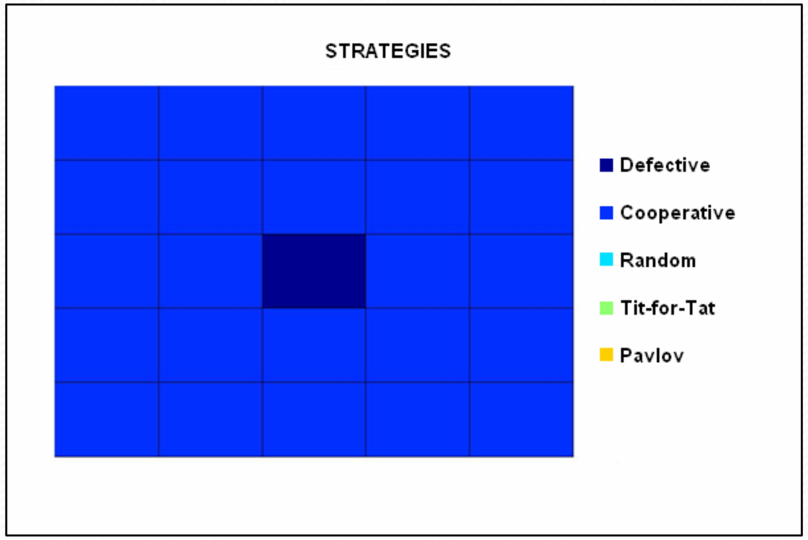
SIMULATION RESULTS

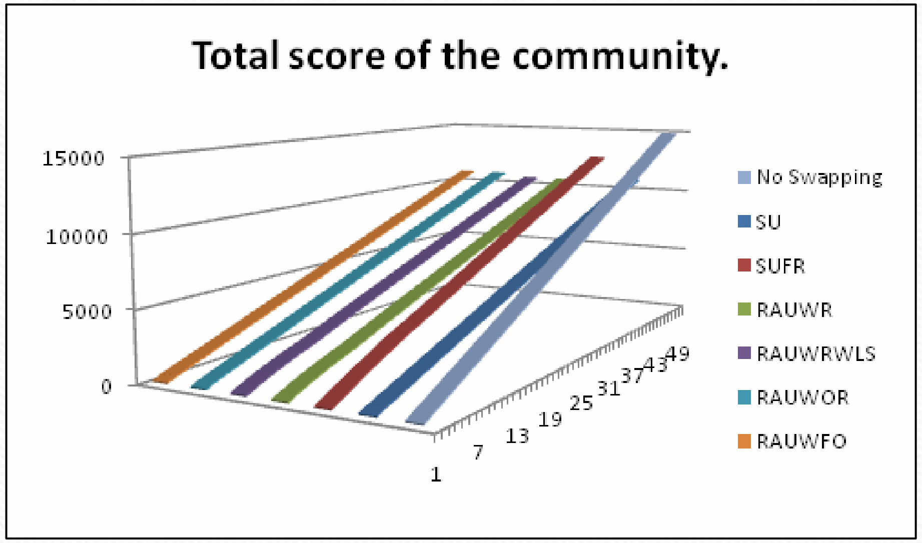
- Simulation results with different original layouts of the strategies of cores and different swapping scenarios, will be presented.
- All communities will be constituted by twenty-five cores and the original layout of the strategy of each core will be illustrated as shown in the following Table.
- Each game starts with the same original layout of strategies and involves all the swapping scenarios. All swapping scenarios occur for fifty rounds.

Strategy	Code and color of the player cell
Defective	🟦 «1»
Cooperative	🟦 «2»
Random	🟦 «3»
Tit-for-Tat	🟩 «4»
Pavlov	🟨 «5»

SIMULATION RESULTS

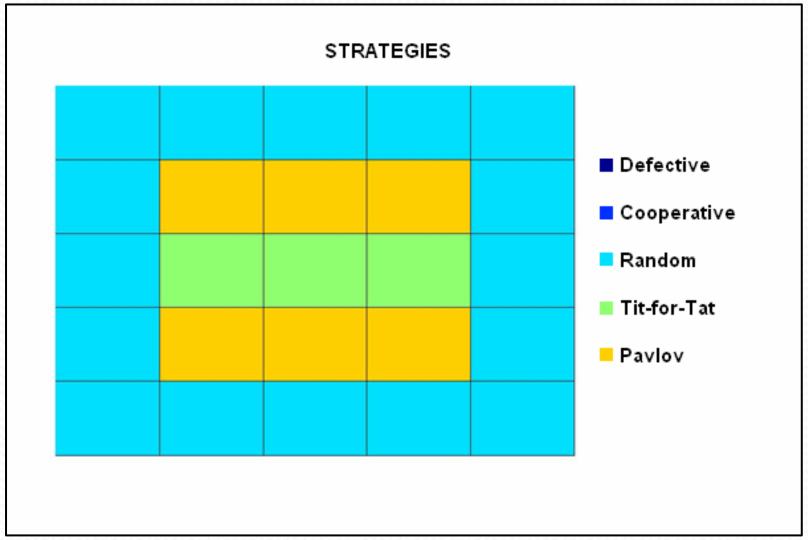
- For the first game a cooperative community was used, with 24 cooperative players and one defective in the center of the community.
- The choice of this pattern of strategies is made to show how defective players react in cooperative communities.

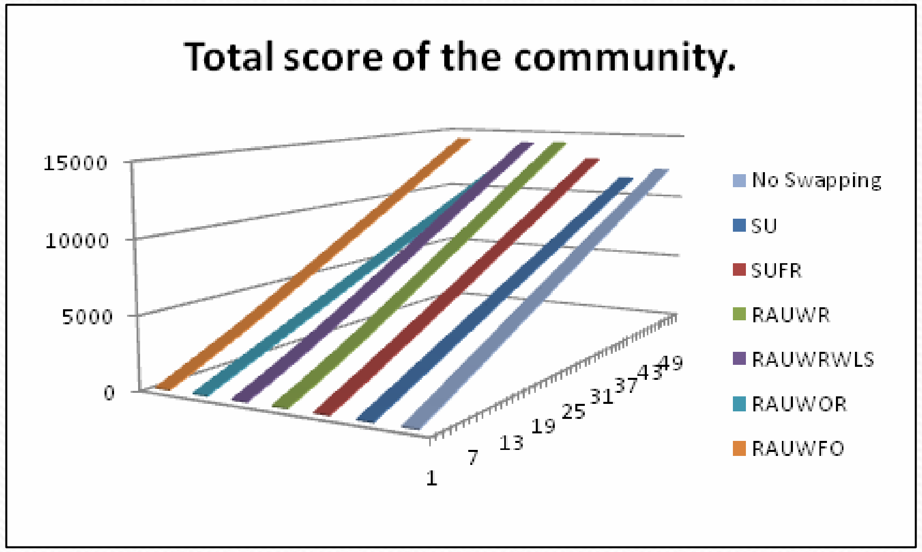




SIMULATION RESULTS

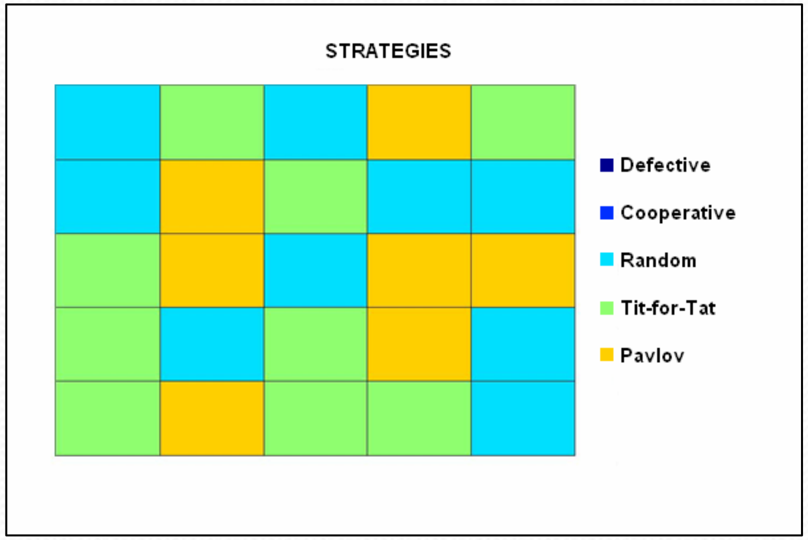
- For the second game a dynamic community was used, with sixteen random players, three Tit-for-Tat and six Pavlov.
- The choice of this pattern of strategies is made to show how dynamic communities, with a fixed pattern, react.

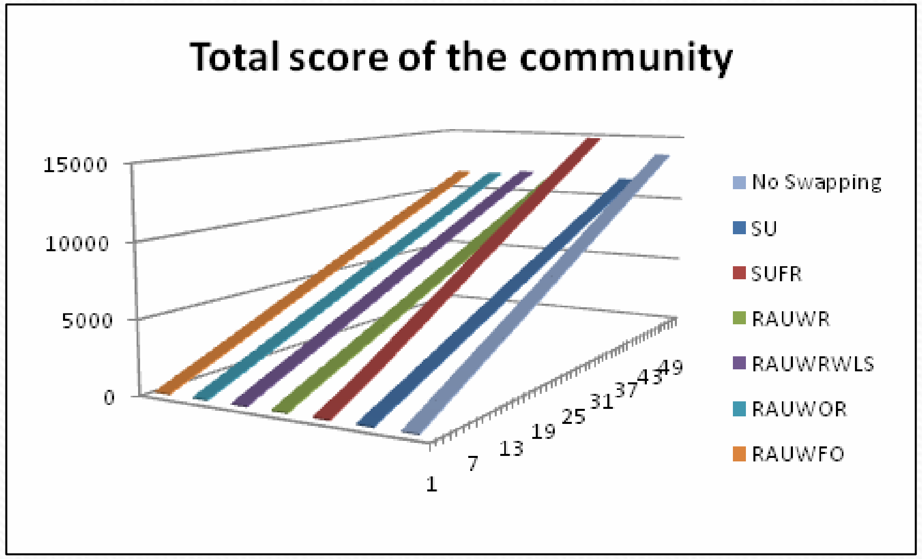




SIMULATION RESULTS

- For the last game a dynamic community was used, with nine random players, nine Tit-for-Tat and six Pavlov.
- The choice of this pattern of strategies is made to show how dynamic communities with a random pattern react.

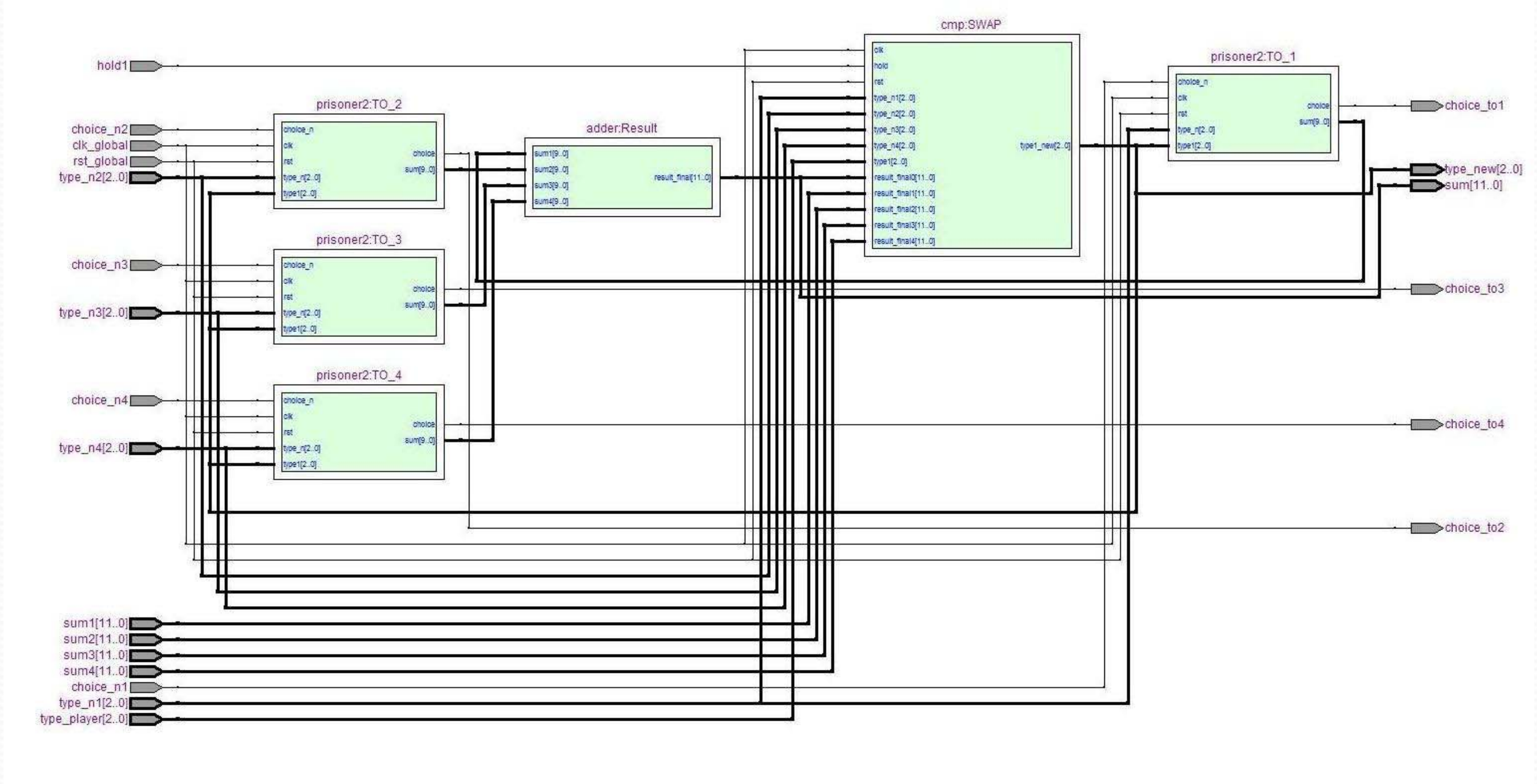




HARDWARE IMPLEMENTATION

- In terms of circuit design and layout, ease of mask generation, silicon-area utilization and maximization of achievable clock speed CA are perhaps the computational structures best suited for a fully parallel hardware realization.
- In contrast to the serial computers, the implementation of the model is motivated by parallelism, an inherent feature of CA that contributes to further acceleration of the model’s operation.
- The hardware implementation of the presented model is based on FPGA logic.
- In order to prove that the hardware implemented system produces the same results with the simulation described in the previous section, its output with certain inputs will be illustrated.

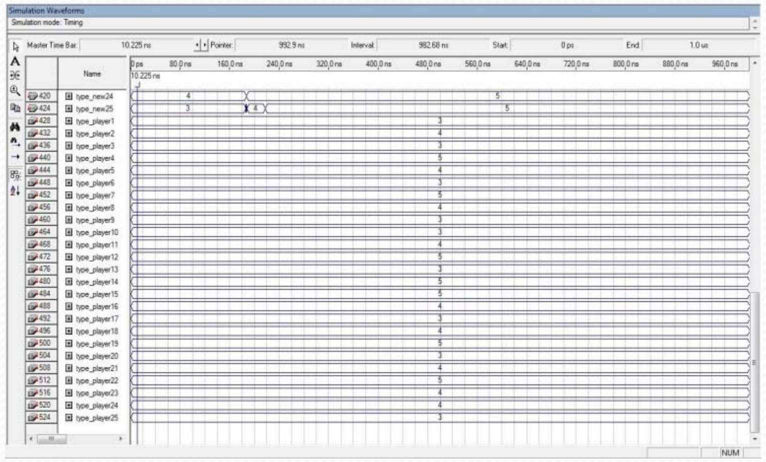
HARDWARE IMPLEMENTATION

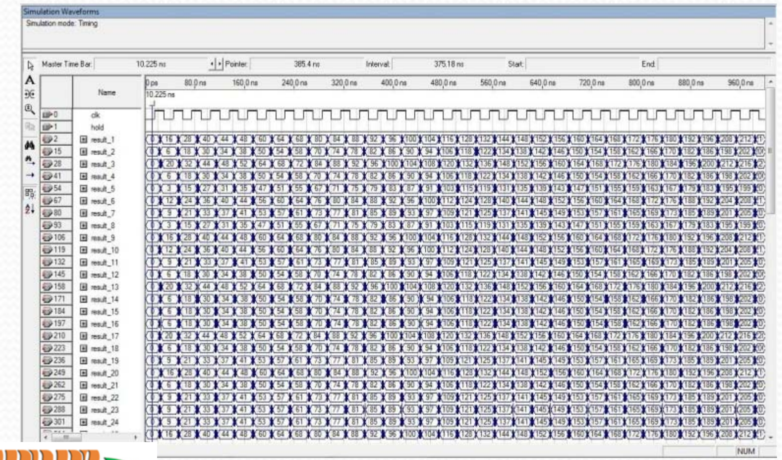


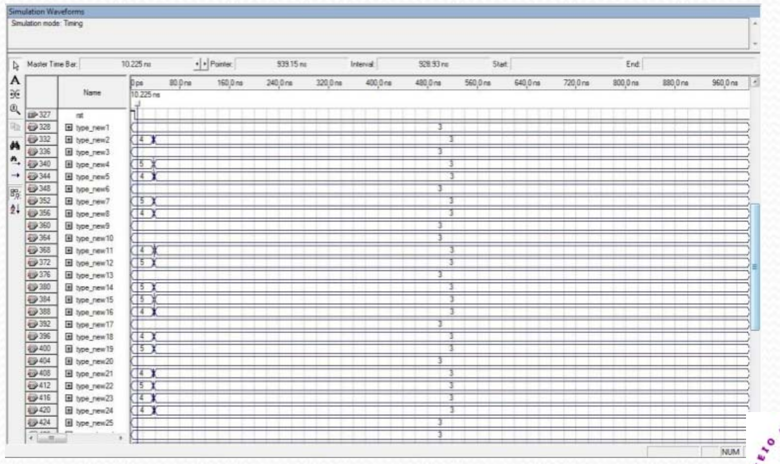
HARDWARE IMPLEMENTATION

The initial layout of the strategies of the third game will be used.

The down left figure shows the system output regarding the sum of payoffs acquired by every player and the down right figure the system output regarding the type of strategy that will be followed on the next round by every player under the SU scenario.

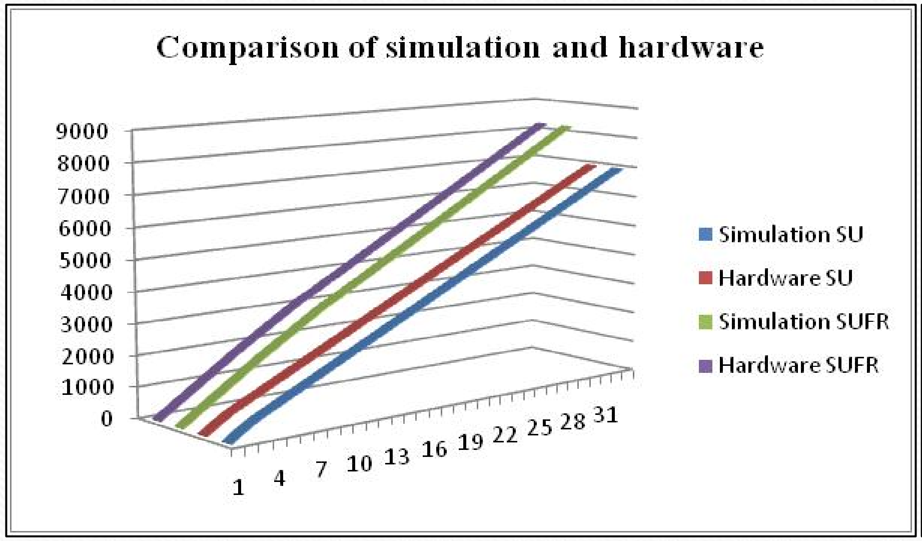


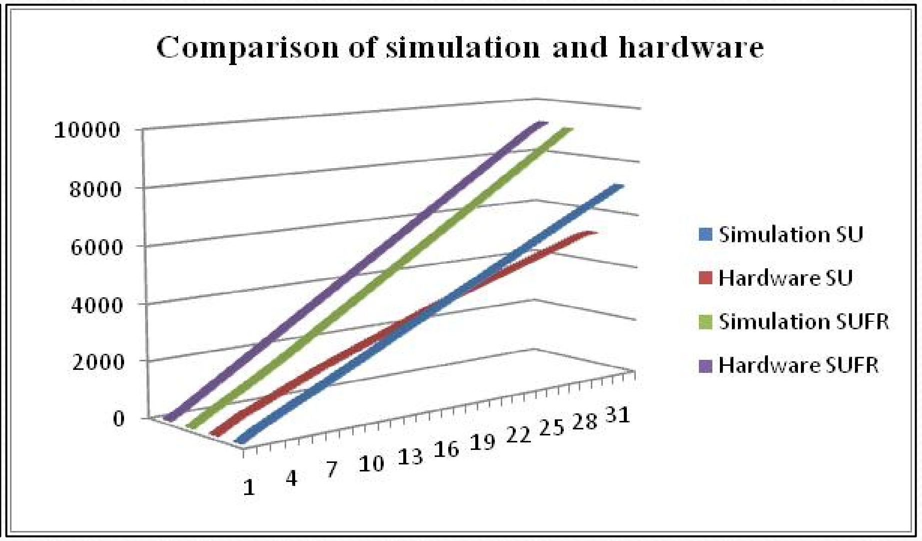




HARDWARE IMPLEMENTATION

- The following figures illustrate the correspondence between the simulation results and the ones produced by the hardware implemented system.
- It is obvious, that for the first game the results are identical.
- On the other hand, the results obtained by the third game are not identical, because of different principles used by the random strategy ("defective" random strategy in hardware).
- However, the improvement using SUFR instead of SU clearly perseveres.





CONCLUSIONS

- The concept of the on-chip memory re-distribution on multicore processors using games on CAs was presented.
- The iterated spatial prisoner's dilemma was introduced as final CA evolution rule.
- From the simulation results it was shown that one of the most important factor is the type of strategies which are included in the community and the location of each strategy in it.
- Also, is proved that a community with random strategy throughout it has a very poor performance, while the random is the most realistic strategy.
- Finally, a FPGA device was developed in order to prove that the concept can easily be automatically designed and attached as a single circuit, real-time, utility in a modern multicore processor.

FUTURE WORK

- Real-life benchmarks are rare and difficult to obtain – especially benchmarks involving more than 8 cores. As a result the method of using different, single core, real-life benchmarks to extract results will be the subject of future work.
- The proposed model is going to be properly enriched with more specific hardware architecture concepts in correspondence to cache coherence found in multicore processors.
- Furthermore, some more technical details regarding the on-chip memory usage in multicore processors are going to be also taken into account, while the strategies could be also differed.
- The payoffs could also become proportional to the resources and the corresponding values that emerge for the under study cores.
- Different neighborhoods will be considered.

The End…
Thank You!