

# АППАРАТНАЯ РЕАЛИЗАЦИЯ СИСТЕМЫ ПОДДЕРЖКИ ПРИНЯТИЯ РЕШЕНИЙ

## HARDWARE IMPLEMENTATION OF A DECISION SUPPORT SYSTEM

V. Amosov  
A. Petrov  
I. Shulgin

*Summary.* In this study, the possibility of a hardware implementation of a Decision Support System based on a tournament decision-making mechanism is demonstrated. A Verilog program for the DSS was developed according to the tournament mechanism's algorithm. The program was tested using manual testing, and the feasibility of configuring the hardware implementation on Cyclone V FPGAs was verified. Additionally, the application of the DSS in comparing game engines to select the best one was examined.

*Keywords:* Decision Support System (DSS), Verilog, Tournament mechanism, FPGA, Quartus, Cyclone V, Hardware implementation, Game engines, Matrix evaluation, Preference weights.

**Амосов Владимир Владимирович**

к.т.н., доцент, Санкт-Петербургский политехнический университет Петра Великого  
amosov\_vv@spbstu.ru

**Петров Александр Владимирович**

Старший преподаватель, Санкт-Петербургский политехнический университет Петра Великого  
petrov\_av@spbstu.ru

**Шульгин Илья Константинович**

Санкт-Петербургский политехнический университет Петра Великого  
shulgin.ik@edu.spbstu.ru

*Аннотация.* В настоящем исследовании показана возможность аппаратной реализации СППР на основе турнирного механизма принятия решений. По алгоритму работы турнирного механизма написана Verilog-программа СППР. Программа оттестирована с применением ручного тестирования, проверена возможность конфигурирования аппаратной реализации в ПЛИС семейства Cyclone V, а также рассмотрено применение СППР при сравнении игровых движков и выбора лучшего.

*Ключевые слова:* Система поддержки принятия решений (СППР), Verilog, Турнирный механизм, ПЛИС, Quartus, Cyclone V, Аппаратная реализация, Игровые движки, Матричные оценки, Веса предпочтений.

## Введение

В данном исследовании представлен процесс разработки и тестирования системы поддержки принятия решений (СППР) [1], реализованной на языке Verilog с использованием турнирного механизма. Представлены фрагменты кода на Verilog, описан алгоритм взаимодействия с программой посредством САПР Quartus, а также результаты её работы. Проведена проверка аппаратной реализации СППР на ПЛИС семейства Cyclone V. Рассмотрено использование данной СППР для сравнения игровых движков и выбора наиболее подходящего.

## Постановка общей проблемы

На сегодня в мире появилось много изделий, которые имеют ограниченный набор функциональностей [2, 3]. Для их создания нет необходимости в программной реализации алгоритмов на универсальных процессорах, способных исполнять любые алгоритмы, так как это дорогостоящий подход [4, 8]. Вместо этого целесообразно реализовать данные алгоритмы напрямую в аппаратуре с помощью ПЛИС и САПР [7]. В данной работе демонстрируется такой метод реализации.

## Постановка задач

Целью данной работы является демонстрация возможности аппаратной реализации СППР, базирующейся на турнирном механизме принятия решений. Для достижения цели необходимо разработать Verilog-модуль СППР согласно алгоритму турнирного механизма [9], провести его тестирование с использованием ручного метода, проверить возможность конфигурирования в ПЛИС, и рассмотреть применение полученного СППР [5].

Для реализации СППР на языке Verilog HDL необходимо:

1. Реализовать автоматическое формирование матриц битового порядка (БО) на основе «таблицы 1», отображающей оценки вариантов решений для каждого БО в диапазоне от 0 до N, где N — это любое целое число.
2. Все массивы в СППР должны быть динамическими, то есть векторами с параметризованными длинами, чтобы поддерживать изменения данных без изменения кода.
3. Начальные данные включают количество вариантов решений, количество БО (предпочтений), веса для каждого БО и таблицу оценок для каждого БО.

4. Каждому БО присваивается признак на основе предпочтений: для пары решений определяется, какой из вариантов предпочтительнее. Если вариант с меньшим значением предпочтительнее, то устанавливается признак False, если наоборот — True.

**Алгоритм работы турнирного механизма**

Пример исходных данных для турнирного механизма представлен в «таблице 1» оценки вариантов решения для каждого предпочтения. В дальнейшем эти данные будут использованы при тестировании. Для каждого БО каждый вариант представлен как реальными данные, так и числами, соответствующими этим данным в смысле соотношений больше-меньше-равно. При качественном подходе ПР важны соотношения, а не реальные данные. Во многих случаях ПР речь о предпочтениях может идти исходя только из соотношений, точных данных может и не быть.

Таблица 1.

Оценка вариантов решения для каждого предпочтения

	Цена (Fals)		Год выпуска (True)		Пробег (Fals)	
LADA Priora 1	143000	5	2008	8	170000	9
LADA Priora 2	150000	10	2009	9	140000	6
LADA Priora 3	148000	8	2009	9	150000	7

Здесь три варианта решения: LADA Priora 1 (1 вариант), LADA Priora 2 (2 вариант), LADA Priora 3 (3 вариант).

Три предпочтения или БО: Цена, Год выпуска, Пробег.

Весовые коэффициенты предпочтений: 30 для цены, 30 для года выпуска и 40 для пробега (сумма коэффициентов в случае аппаратной реализации должна равняться 100).

Цель принятия решения: выбрать лучший вариант, исходя из представленных предпочтений.

Для каждого предпочтения, то есть для каждого столбца в таблице оценки вариантов решения, строится матрица бинарных отношений (БО)  $||ij||$  (номер строки  $i$  — первый вариант решения в сравнении со вторым вариантом  $j$  — это номер столбца), элемент матрицы БО  $(i,j)$  может иметь два значения: 1 — если оценка первого предпочтительней оценки второго или они совпадают, и 0 — если оценка второго предпочтительней оценки первого. Для СППР делается допущение, что все БО (предпочтения) слабополные или слабосвязные, то есть оцениваются все пары хотя бы в одну сторону, кроме оценки варианта самого с собой. Это значит, что диагональные элементы матриц БО не рассматриваются. Изолированных, неохваченных вариантов в данном рассмотрении нет.

Матрицы БО выглядят следующим образом, здесь 1, 2, 3 — три варианта:

Таблица 2.

Попарная оценка вариантов по предпочтению цена

Цена	1	2	3
1	1	1	1
2	0	1	0
3	0	1	1

Таблица 3.

Попарная оценка вариантов по предпочтению год выпуска

Год выпуска	1	2	3
1	1	0	0
2	1	1	1
3	1	1	1

Таблица 4.

Попарная оценка вариантов по предпочтению пробег

Пробег	1	2	3
1	1	0	0
2	1	1	1
3	1	0	1

Затем в соответствии с турнирным механизмом для каждого из предпочтений совершается суммирование по строчкам матрицы БО, и варианты получают баллы за единицы в соответствующих им строчках, помноженные на соответствующий каждому предпочтению весовой коэффициент. В случае если два сравниваемых варианта имели одинаковую оценку по предпочтению и, следовательно, единицы на симметричных позициях в матрице БО, то в этом случае балл делится пополам.

Для каждого из вариантов решения баллы, полученные для каждого предпочтения умноженные на весовой коэффициент, суммируются, получаем столбец сумм для каждого варианта и в результате по сумме баллов определяются места ранжирования вариантов решения. Лучшим вариантом будет набравший наибольшую сумму.

**Вычисление результатов турнирного механизма**

Определим сколько раз каждый вариант решения предпочтительней других по каждому БО.

1 вариант: два раза только по цене (весовой коэффициент 30), следовательно  $30+30=60$ .

2 вариант: по Году выпуска (весовой коэффициент 30) предпочтительней 1 варианта, следовательно 30 и симметричен со 2 вариантом, следовательно 15; по Пробегу (весовой коэффициент 40) предпочтительней 2 раза, следовательно  $40+40=80$ . Всего:  $30+15+40+40=125$ .

3 вариант: по Цене (весовой коэффициент 30) предпочтительней 1 раз, следовательно 30; по Году выпуска (весовой коэффициент 30) предпочтительней 1 раз, следовательно 30 и симметричен 1 раз, следовательно 15; по Пробегу (весовой коэффициент 40) предпочтительней 1 раз, следовательно 40. Всего:  $30+30+15+40=115$

### Аппаратная реализация СППР

В аппаратной реализации СППР на основе турнирно-го механизма были введены следующие упрощения [6].

При подаче сигнала *analyse*, обрабатываются данные только об одном предпочтении, то есть для обработки данных по трём предпочтениям потребуется три раза подавать сигнал *analyse* и три раза изменять данные о предпочтении (оценки и весовой коэффициент). Это обусловлено экономией ширины входных портов.

Вместо построения матриц БО программа сразу подсчитывает баллы для каждого из вариантов. Это позволяет упростить логику работы и сэкономить ресурсы на внутреннюю память.

В качестве семейства устройств было выбрано семейство Cyclone V. Аппаратная реализация позволяет обрабатывать данные, но не более семи вариантов. Исходный код представлен в Приложении 1.

### Тестирование Verilog-программы СППР

В работе для проверки правильности результатов Verilog-программы СППР было применено ручное тестирование.

При заданных исходных данных (таблица 1) были вручную посчитаны баллы для всех трёх вариантов с учётом весовых коэффициентов и получены результаты: для LADA Priora 1 — 60, для LADA Priora 2 — 125, для LADA Priora 3 — 115.

Ввод исходных данных при работе программы СППР представлен на рисунке 1:

Результаты работы программы представлены на рисунке 2:

В результате симуляции наибольшее количество баллов, а именно 125, набрала **LADA Priora 2**, она занимает первое место. Далее по уменьшению баллов идут LADA Priora 3 (115 баллов) и LADA Priora 1 (60 баллов).

Полученные в результате работы программы баллы для каждого предпочтения совпадают с посчитанными вручную. На основании этого делаем вывод о корректной работе программы.

### Результаты проверки аппаратной реализации СППР на ПЛИС семейства Cyclon V

Проверка проводилась с помощью САПР Quartus и показала положительный результат, был получен отчёт о компиляции, а также RTL-схема и технологическая схема размещения программы в ПЛИС семейства Cyclon V.

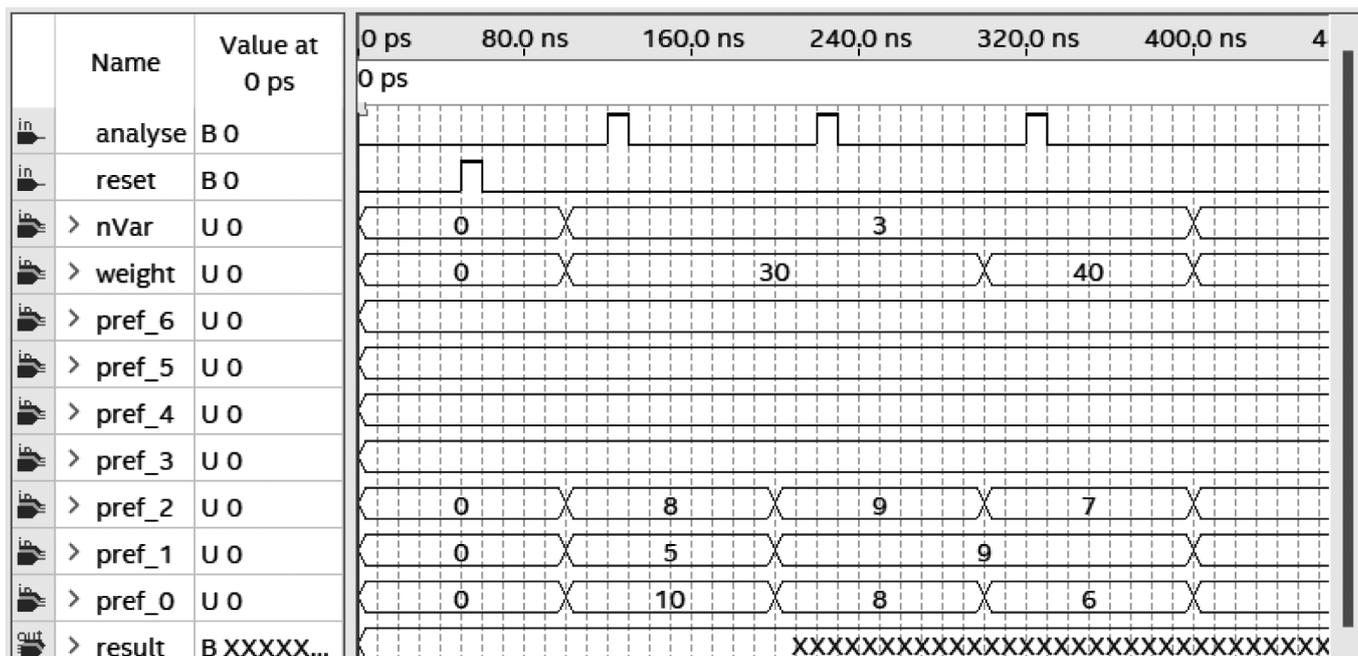


Рис. 1. Исходные данные

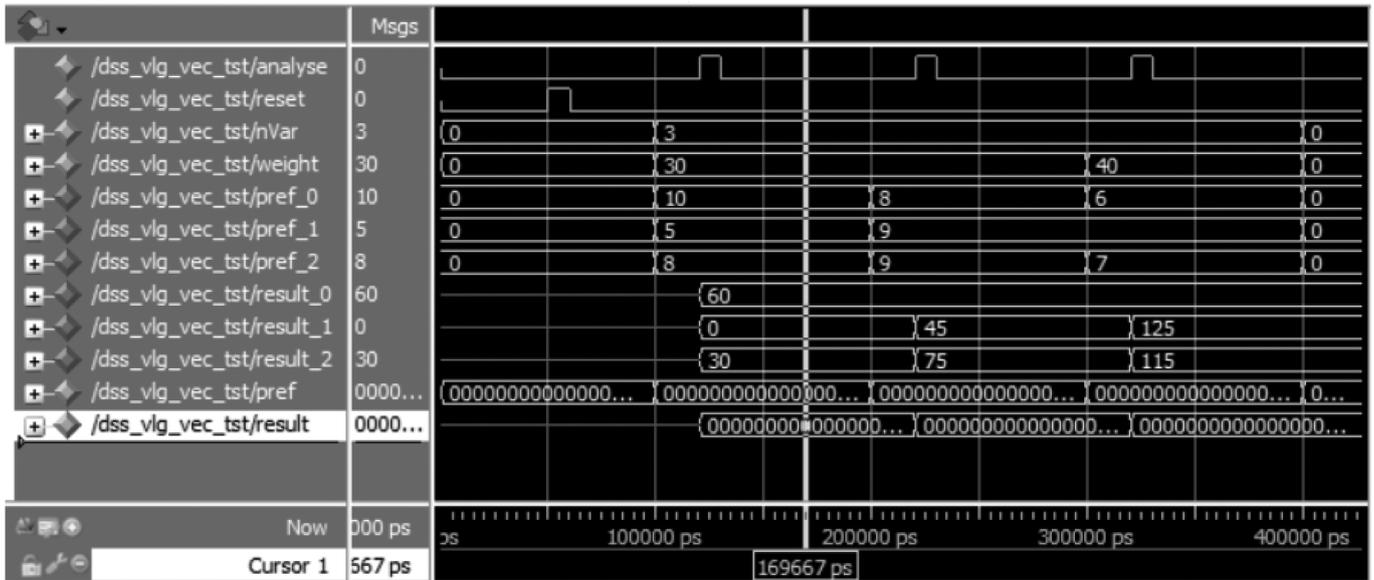


Рис. 2. Результаты симуляции

**Применение СППР для сравнения игровых движков**

В качестве вариантов выбраны наиболее популярные игровые движки, они позволяют создавать различные продукты, от небольших платформеров и визуальных новелл до крупных AAA-проектов.

В сравнении используются только открытые игровые движки, но многие компании для своих проектов нередко создают собственные закрытые движки, которые не подходят для создания новых проектов и не доступны большинству разработчиков.

В таблице 5 ниже представлены участвующие в сравнении игровые движки и их субъективные оценки по предпочтениям.

Ввод исходных данных при работе программы СППР представлен на рисунке 3:

Результаты работы программы представлены на рисунке 4:

В результате лидером по субъективным критериям и оценкам оказался **Godot**, набравший **295** баллов, это говорит о том, что данный движок отлично подходит для небольших проектов и дружелюбен к разработчику. Второе место занял **Unreal Engine**, а третье — **Unity**, что подтверждает их заслуженную популярность. Движки **Ren'Py** и **Gamemaker Studio 2** недалеко отстали от конкурентов, что говорит о том, что они являются востребованными в своей нише. **CryEngine** занял последнее место, набрав меньше всех баллов — **150**. Возможно, это следствие субъективности моих оценок и необъективности предпочтений, ведь **CryEngine** подходит для соз-

Таблица 5.

Оценки по предпочтениям для игровых движков

Весовой коэффициент	Предпочтение	Варианты					
		Unity	Unreal Engine	CryEngine	Godot	Ren'Py	Gamemaker Studio 2
15	Документация	9	8	6	10	7	9
15	Сообщество	9	9	5	8	7	6
10	Удобство	5	7	6	8	9	9
20	Требования к навыкам разработчика	5	4	4	7	6	8
20	Графика	7	8	10	5	6	3
10	Поддержка передовых технологий	7	9	10	5	0	0
10	Стоимость	8	9	7	10	10	8

дания крупных высокопроизводительных приложений, в то время как **Godot** для менее масштабных и менее требовательных проектов.

**Приложение 1. Verilog — код программы СППР**

```

module dss #(
parameter MAXNVAR = 7,
parameter MAXNVARB = 3
)
(
input [MAXNVARB-1:0] nVar,

```

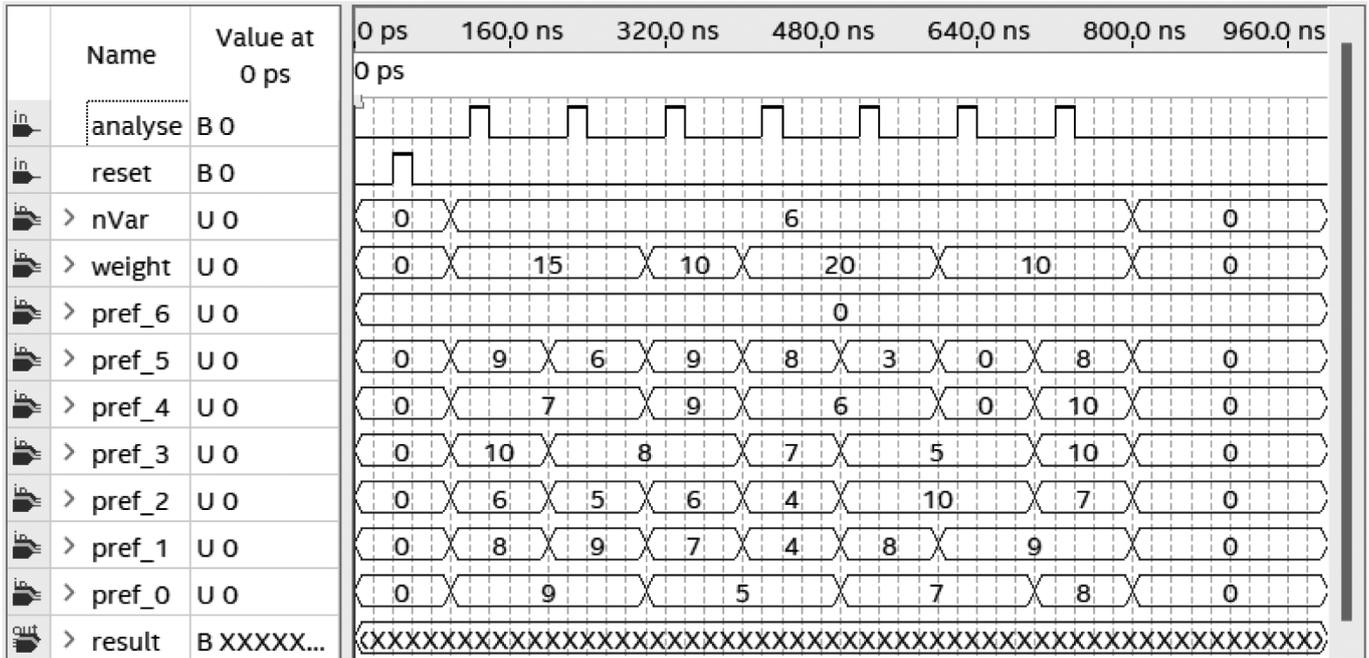


Рис. 3. Исходные данные

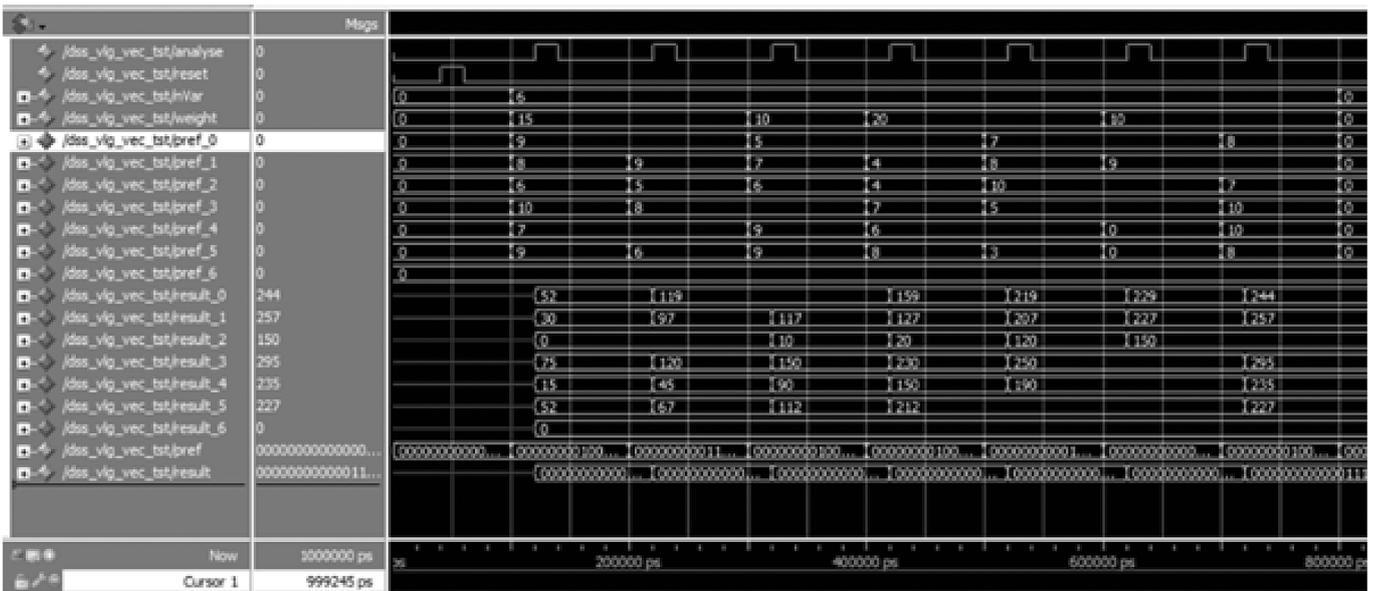


Рис. 4. Результаты симуляции

```
input [MAXNVAR-1:0] weight,
input [(MAXNVAR+1)*4-1:0] pref,
input analyse,
input reset,
output reg [MAXNVAR*10-1:0] result
);

reg [9:0] scores [MAXNVAR-1:0];
integer i, j;

always @(posedge reset or posedge analyse) begin
if (reset) begin
// Используем generate для сброса больших массивов
```

```
for (i = 0; i < MAXNVAR; i = i + 1) begin
scores[i] <= 0;
end
end else if (analyse) begin
// Обнуление счетчиков перед новым анализом
for (i = 0; i < MAXNVAR; i = i + 1) begin
scores[i] <= 0;
end

for (i = 0; i < nVar; i = i + 1) begin
for (j = 0; j < nVar; j = j + 1) begin
if (i != j) begin
// Извлекаем предпочтения один раз
```

```

reg [3:0] pref_i = pref[i*4 +: 4];
reg [3:0] pref_j = pref[j*4 +: 4];

if (pref_i != 0 && pref_j != 0) begin
// Уменьшение обращения к массиву на каждом шаге
if (pref_i > pref_j) begin
scores[i] <= scores[i] + weight;
end else if (pref_i == pref_j) begin
scores[i] <= scores[i] + (weight >> 1);
end
end
end
end
end

// Обновляем результаты после формирования всех
очков
for (i = 0; i < MAXNVAR; i = i + 1) begin
result[i*10 +: 10] <= scores[i];
end
end
end
endmodule

```

### Заключение

В статье продемонстрирована возможность аппаратной реализации алгоритмов с ограниченной функцио-

нальностью, избавляющей от необходимости использовать универсальные процессоры для программной реализации любого алгоритма. Описаны инструменты и методология аппаратной реализации алгоритмов.

Показана возможность параллелизации выполнения сложных программных систем, состоящих из множества алгоритмов, часть из которых исполняется аппаратно, а часть — программно [10, 11].

Описана разработка, исследование и тестирование Verilog-модуля системы поддержки принятия решений (СППР) на основе турнирного механизма. Проведено ручное тестирование: результаты выбора автомобиля, полученные вручную по алгоритму, были сопоставлены с результатами, формализованно введенными в Verilog-модуль СППР. Полученные результаты совпали с расчетными.

С использованием САПР Quartus проведена проверка Verilog-модуля СППР на синтезируемость и реализация его на ПЛИС семейства Cyclone V. Рассмотрено применение Verilog-модуля СППР для сравнения игровых движков с целью выбора оптимального

### ЛИТЕРАТУРА

1. Чечнев В.Б. Использование систем поддержки принятия решений в автоматизации процессов принятия решений // Электронные библиотеки. — 2024. — Т. 27, № 4. — С. 15–25.
2. Воронова Н.С., Яковлева Е.А., Шарич Э.Э., Яковлева Д.Д. Системы поддержки принятия решений в составе интеграционных механизмов финансового рынка ЕАЭС в контексте устойчивого развития // Экономика, предпринимательство и право. — 2021. — Т. 11, № 12. — С. 3105–3120.
3. Бурцев М.В., Поворознюк А.И. Архитектура системы поддержки принятия решений в медицине, основанной на комбинированном решающем правиле // Вестник НТУ «ХПИ». Серия: Информатика и модели управления. — 2020. — № 2. — С. 45–50.
4. Шуршев Т.В., Хоменко Т.В. Разработка системы интеллектуальной поддержки принятия решений на основе сверточной нейронной сети // XXI век: итоги прошлого и проблемы настоящего плюс. — 2024. — Т. 13, № 1 (65). — С. 123–130.
5. Аксенов К.А., Спицина И.А. Метод поддержки принятия решений при разработке информационных систем на основе мультиагентного подхода: монография. — Москва: Издательский дом Академии Естествознания, 2023. — 180 с.
6. Иванов П.П., Сидоров А.А. Аппаратная реализация систем поддержки принятия решений на основе FPGA // Вестник компьютерных и информационных технологий. — 2022. — № 7. — С. 35–42.
7. Петрова Е.С., Стефэнеску Р. Принятие решений, некоторые индивидуальные стили принятия решений и программное обеспечение для принятия решений и стратегического планирования // Przegląd Nauk o Obronności. — 2022. — Т. 1. — С. 1–12.
8. Косников С.Н., Золкин А.Л., Атаева Л.Б., Жильцов С.А. Разработка пользовательского интерфейса и управление информацией в системах поддержки принятия решений // Международный журнал. Естественно-гуманитарные исследования. — 2023. — № 5 (49). — С. 406–409.
9. Рвачева А.С., Арнаев М.А., Никитин Н.Н., Манджиева А.Б., Патдыева А., Красноруцкая К.А., Чимидов Д.Б. Анализ существующих средств ИКТ и программных решений в области поддержки принятия решения коммерческих организаций // Экономика и предпринимательство. — 2023. — № 5. — С. 824–827.
10. Демидовский А.В., Бабкин Э.А. Интегрированные нейросимволические системы поддержки принятия решений: проблемы и перспективы // Бизнес-информатика. — 2021. — Т. 15, № 3. — С. 7–23.
11. Назарова О.О. Эволюция систем поддержки принятия решений // Дневник науки. — 2022. — № 3 (63). — С. 1–7.

© Амосов Владимир Владимирович (amosov\_vv@spbstu.ru); Петров Александр Владимирович (petrov\_av@spbstu.ru);  
Шульгин Илья Константинович (shulgin.ik@edu.spbstu.ru)

Журнал «Современная наука: актуальные проблемы теории и практики»