

Федеральное государственное автономное образовательное учреждение
высшего образования

"Национальный исследовательский университет
"Высшая школа экономики"

Московский институт электроники и математики им. А.Н.Тихонова
Департамент компьютерной инженерии

Курс «Проектирование систем на кристалле»

ОТЧЕТ

о выполнении домашнего задания

Выполнили:

Апьюк В.Р. БИВ214

Воркова В.С. БИВ214

Ляпунова С.А. БИВ214

Запорожан Д. Р. БИВ214

Москва, 2024

Оглавление

Задание.....	3
Выполнение работы.....	4
1. Датасет.....	4
2. Документация разработчика.....	9
3. Документация пользователя.....	11
Выводы.....	14
Список литературы.....	15
Приложения.....	16

Задание

Разработка рекомендательной системы выбора платы для проведения прототипирования модели. Необходимо на основе характеристик отладочных плат ПЛИС, которые есть в лаборатории САПР, реализовать рекомендательную систему для выбора платы для прототипирования проекта в зависимости от аппаратных требований. Для решения этой задачи рекомендуется создать базу данных, в которой будут храниться все характеристики отладочных плат. На вход программы будут подаваться указанию по требуемым характеристикам. На основе этих параметров программа должна будет подобрать наиболее подходящую отладочную плату.

Выполнение работы

1. Датасет

Для выполнения задания требуется создать базу данных, в которой будут храниться все характеристики отладочных плат.

Для базы данных были выбраны платы: DE1-SoC (рис. 1), DE10-Lite (рис. 2), DE10-Standard (рис. 3), DE10-Nano (рис. 4), DE0-CV (рис. 5).

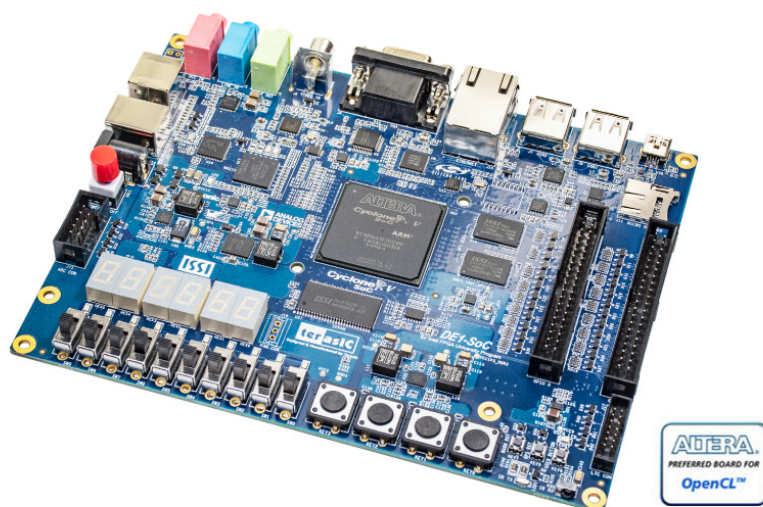


Рисунок 1. DE1-SoC

DE1-SoC представляет собой мощную платформу для проектирования, основанную на микропроцессоре Altera System-on-Chip (SoC) FPGA, который объединяет последние двухъядерные ядра Cortex-A9 для встроенных вычислений с лидирующими по производительности логическими схемами для максимальной гибкости проектирования. Микропроцессор Altera SoC включает в себя жесткий процессорную систему (HPS) на основе ARM, состоящую из процессора, периферийных устройств и интерфейсов памяти, которые связаны безотказно с логикой FPGA с помощью высокоскоростной интерконнектной структуры. Плата DE1-SoC включает в себя аппаратное обеспечение, такое как высокоскоростная память DDR3, видео и аудио возможности, сеть Ethernet и многое другое.

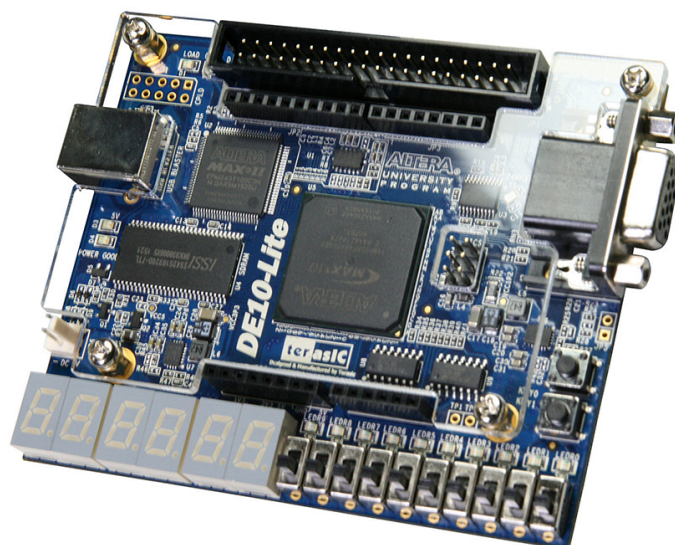


Рисунок 2. DE10-Lite

DE10-Lite - это экономически эффективная платформа на основе Altera MAX 10 FPGA. Плата использует максимальную производительность MAX 10 FPGA, которая имеет приблизительно 50 тысяч логических элементов (LE) и аналого-цифровой конвертер (ADC). Она включает в себя USB-Blaster, SDRAM, акселерометр, VGA-вывод, 2x20-разрядный разъем для расширения GPIO и разъем для расширения Arduino UNO R3 в компактном размере. Комплект предоставляет идеальное решение для прототипирования на уровне системы для промышленного, автомобильного, потребительского и многих других рыночных приложений.

Кроме того, комплект DE10-Lite содержит множество примеров проектов и программных инструментов для пользователей, чтобы легко развивать свои приложения на основе этих ресурсов проектирования.

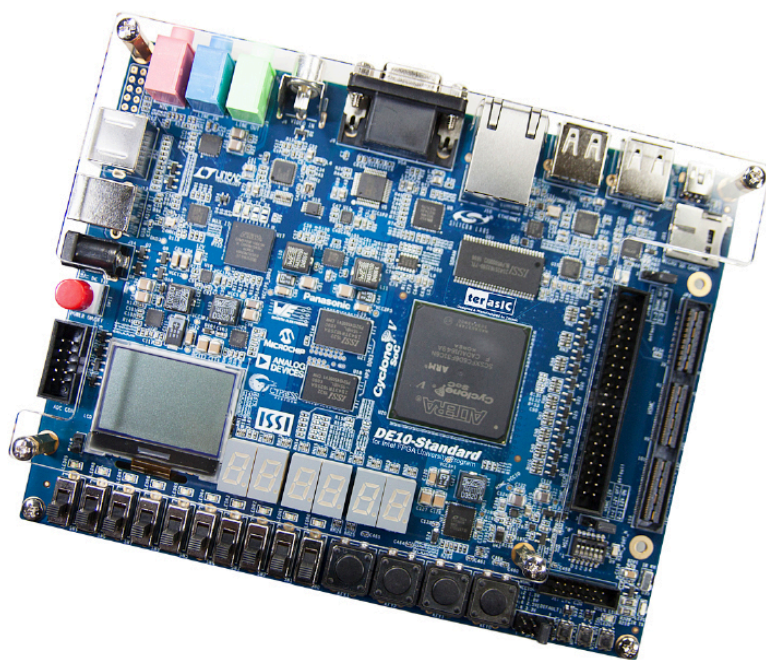


Рисунок 3. DE10-Standard

Набор для разработки DE10-Standard представляет собой надежную аппаратную платформу, построенную вокруг FPGA системы на кристалле (SoC) от Intel, которая сочетает в себе последние двухъядерные встроенные ядра Cortex-A9 с передовой программируемой логикой для обеспечения максимальной гибкости дизайна. Теперь пользователи могут использовать мощь огромной перенастраиваемости в сочетании с высокопроизводительной системой процессора с низким энергопотреблением. SoC от Altera интегрирует жесткую процессорную систему (HPS) на базе ARM, состоящую из процессора, периферийных устройств и интерфейсов памяти, которые безупречно связаны с тканью FPGA с помощью магистрали высокой пропускной способности. Разработочная плата DE10-Standard включает такое оборудование, как память DDR3 высокой скорости, возможности видео и аудио, сетевое подключение Ethernet и многое другое.

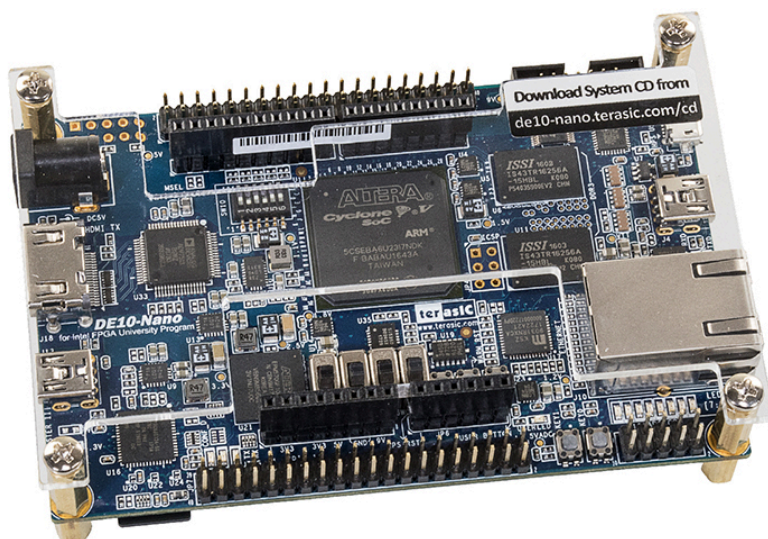


Рисунок 4. DE10-Nano

Разработочная плата DE10-Nano открывает перед пользователями широкие возможности для создания разнообразных электронных проектов. Благодаря встроенной FPGA системе на кристалле от Intel, эта плата предоставляет исключительную гибкость в дизайне и программировании. С двухъядерными ядрами Cortex-A9 и высокоскоростной памятью DDR3, DE10-Nano идеально подходит для сложных мультимедийных задач и обещает превосходную производительность. Интегрированный USB-Blaster II, SDRAM и разъемы расширения делают её универсальным инструментом для разработчиков, а поддержка операционных систем Windows XP и новее гарантирует легкость в использовании. С такими характеристиками, как 12-битный АЦП и мощная программируемая логика, DE10-Nano становится незаменимым ресурсом для инновационных электронных проектов.

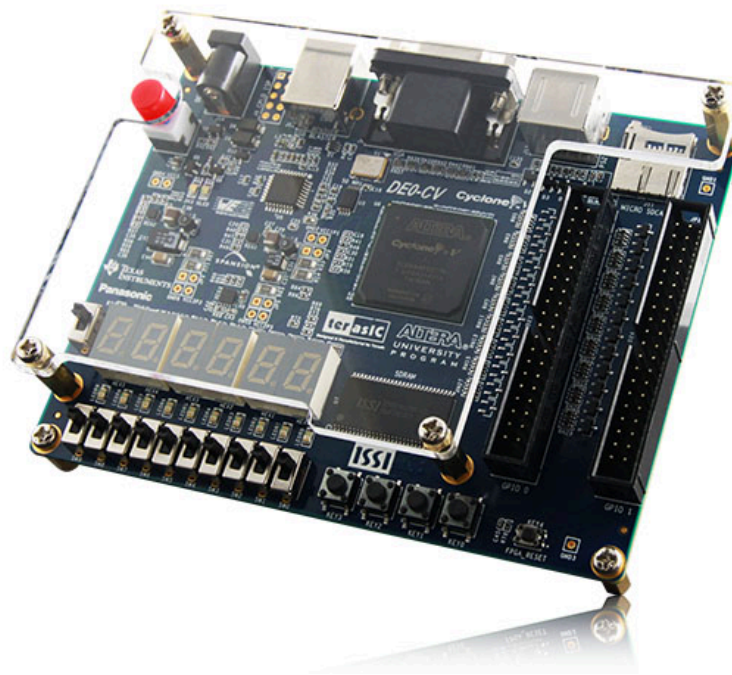


Рисунок 5. DE0-CV

Плата DE0-CV — это универсальное устройство, предназначенное для реализации множества электронных проектов, от базовых схем до сложных мультимедийных приложений. Основой платы служит мощный FPGA Cyclone V от Altera, который обеспечивает высокую производительность и гибкость в проектировании. Плата оснащена всем необходимым для создания прототипов: от USB Blaster для программирования и конфигурирования до SDRAM и интерфейсов для видео. DE0-CV идеально подходит для образовательных целей, разработки прототипов и создания мостовых схем. Её компактные размеры и богатый набор функций делают её предпочтительным выбором для инженеров и хоббистов, желающих экспериментировать с FPGA и разрабатывать свои собственные устройства. С Cyclone V 5CEBA4F23C7N, 49К логических элементов и 64 МБ SDRAM, DE0-CV предлагает отличную платформу для обучения и инноваций в области цифровой электроники.

2. Документация разработчика

Программа представляет собой интерактивное приложение на Python, использующее библиотеку `ipywidgets` для создания пользовательского интерфейса в Jupyter Notebook. Вот общая суть программы:

1. **Установка и импорт библиотек:** Сначала устанавливаются необходимые библиотеки (`ipywidgets`), а затем импортируются библиотеки `pandas` и `ipywidgets` для работы с данными и создания интерфейса.
2. **Чтение данных:** Программа читает данные из файла Excel (`data.xlsx`) и загружает их в `DataFrame df`. Предполагается, что файл находится в рабочей директории.
3. **Обработка данных:**
 - Из `DataFrame` удаляются колонки 'Название', 'Ссылка', 'Тип ПЛИС', и оставшиеся колонки сохраняются в переменной `characteristics`.
 - Все столбцы, начинающиеся с "Наличие", преобразуются так, что значения 1 заменяются на "да", а 0 на "нет".
 - Значения "-", "nan" и другие нежелательные значения заменяются на "неважно".
4. **Создание виджетов:**
 - Для каждой характеристики из `characteristics` создается выпадающий список (`Dropdown`) с уникальными значениями этой характеристики и дополнительным значением "неважно".
 - Каждому выпадающему списку соответствует метка (`Label`), отображающая название характеристики.
5. **Функция поиска подходящей платы:**

- Определена функция `find_best_board`, которая оценивает каждую плату в `DataFrame` на основе выбора пользователя в выпадающих списках.
 - Если выбор пользователя совпадает с характеристикой платы, то плата получает балл.
 - В конце функция выводит название платы с наибольшим количеством баллов.
6. **Кнопка для запуска поиска:** Создается кнопка (`Button`), которая при нажатии вызывает функцию `find_best_board`.
7. **Отображение виджетов и кнопки:** Все виджеты и метки собираются в вертикальный контейнер (`VBox`) и отображаются, а кнопка отображается отдельно под виджетами.

В итоге, пользователь может выбрать характеристики интересующей его платы через выпадающие списки, и после завершения выбора нажать на кнопку, чтобы увидеть наиболее подходящую плату на основе его предпочтений. Это удобный способ фильтрации данных и поиска оптимальных решений в интерактивной форме.

3. Документация пользователя

Шаги использования приложения:

1. Выбор характеристик

- После запуска программы отображается ряд выпадающих списков, каждый из которых соответствует определенной характеристике платы, такой как тип памяти, количество портов, поддержка интерфейсов и т.д.
- Выберите из каждого списка ту характеристику, которая важна для вас (рис. 6).

Диапазон аналогового входа АЦП	0 ~ 4.096 V
PLLs HPS	0
PLLs ПЛИС	6
Аудио	24-bit CODEC, Line-in, line-out, microphone-in jacks
Видеовход	TV Decoder (NTSC/PAL/SECAM) and TV-in connector
Датчики	G-Sensor on HPS
Дисплей	24-bit VGA DAC
Кнопки	6 (4 FPGA; 2 HPS Reset Buttons (HPS_RST_n and HPS_WARM_RST_n))
Количество каналов АЦП	8
Количество логических элементов ПЛИС	85000
Наличие коммуникации IR Emitter/Receiver	да
Наличие коммуникации PS/2 mouse/keyboard	да
Наличие коммуникации Ethernet	10/100/1000
Наличие коммуникации USB	USB 2.0 Host (2 порта), UART to USB
Наличие разъема One HSMC Connector(Configurable I/O Standards 1.5/1.8/2.5/3.3V)	нет
Наличие разъема Arduino Uno R3 Connector, including six ADC channels.	нет
Наличие разъема One 10-pin ADC Input Header	да
Наличие разъема One 2x20 GPIO Connector(voltage levels: 3.3V)	нет
Наличие разъема One LTC connector (One Serial Peripheral Interface (SPI) Master ,one I2C and one ...	да
Наличие разъема Two 40-pin Expansion Headers	да
Объем внешней памяти HPS	1GB (2x256Mx16) DDR3 SDRAM
Объем внешней памяти ПЛИС	64MB (32Mx16) SDRAM
Объем встроенной памяти ПЛИС (Kbits)	4.45
Переключатели	10
Питание	12V DC
Процессор	Dual-core ARM Cortex-A9 (HPS)
Разрешение АЦП	12 Bit
Светодиоды	11 (FPGA x10 ; HPS x 1)
Семисегментный дисплей	6
Частота дискретизации АЦП	500Ksps
Завершить выбор	

Рисунок 6. Пример выбора характеристик

2. Игнорирование неважных параметров

- Если конкретная характеристика для вас не имеет значения, оставьте в соответствующем выпадающем списке значение “неважно”. Это позволит исключить эту характеристику из поиска (рис. 7).

Диапазон аналогового входа АЦП	неважно ▼
PLLs HPS	неважно ▼
PLLs ПЛИС	неважно ▼
Аудио	24-bit CODEC, Line-in, line-out, microphone-in jacks ▼

Рисунок 7. Пример выбора характеристик

3. Завершение выбора

- После того как вы сделаете свой выбор по всем характеристикам, нажмите кнопку “Завершить выбор” для запуска процесса подбора платы (рис. 8).

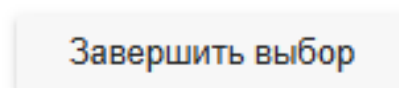


Рисунок 8. Кнопка “Завершить выбор”

4. Оценка вариантов

- Приложение автоматически оценит все доступные платы, начисляя баллы за каждую характеристику, которая соответствует вашему выбору.

5. Получение результатов

- По завершении оценки, приложение определит плату с наибольшим количеством баллов и представит её вам как наиболее подходящую вариант (рис. 9).



Рисунок 9. Вывод результата

С помощью данного приложения вы сможете быстро и эффективно отфильтровать множество вариантов, чтобы найти идеальную плату, соответствующую вашим спецификациям. Это упрощает процесс выбора и помогает вам принять обоснованное решение.

Выводы

В рамках поставленной задачи была разработана рекомендательная система, предназначенная для подбора отладочных плат ПЛИС в лаборатории САПР. Система учитывает аппаратные требования проекта и специфические параметры для проектов сети на кристалле. В качестве основы для системы создана база данных, содержащая характеристики различных плат, доступных в лаборатории. Интерактивное приложение, реализованное на Python с использованием библиотеки `ipywidgets`, позволяет пользователям визуально выбирать необходимые характеристики через удобный пользовательский интерфейс в Jupyter Notebook. Функция `'find_best_board'` анализирует выбор пользователя и выдает рекомендацию по наиболее подходящей плате. Результаты работы программы демонстрируют, что разработанная система обеспечивает точный и целенаправленный подбор платы, соответствующей требованиям конкретного проекта, что значительно упрощает процесс выбора и помогает принять обоснованное решение.

Список литературы

1. Terasic - The leading designer and provider of FPGA development kits. – URL: <https://www.terasic.com.tw/cgi-bin/page/archive.pl?Language=English&CategoryNo=163> (дата обращения: 10.06.2024).
2. Разработка и применение систем на кристалле. – URL: <https://publications.hse.ru/mirror/pubs/share/direct/430635780.pdf> (дата обращения: 10.06.2024).
3. Введение в разработку сетей на кристалле (NoC). – URL: <https://habr.com/ru/companies/lanit/articles/421401/> (дата обращения: 10.06.2024).
4. ipywidgets Documentation. – URL: <https://ipywidgets.readthedocs.io/en/latest/> (дата обращения: 10.06.2024).
5. Применение ipywidgets для создания интерактивных приложений в Jupyter Notebook. – URL: <https://habr.com/ru/companies/neoflex/articles/766132/> (дата обращения: 10.06.2024).

Приложения

Приложение 1. Датасет

Название	Ссылка	Тип ПЛИС	Процессор	Количество логических элементов ПЛИС	PLLs ПЛИС	PLLs HPS
De1 SoC	https://www.terasic.com.tw/cgi-bin/page/archive.pl?Language=English&CategoryNo=167&No=836#contents	Cyclone V SoC 5CSEMA5F31C6 Device	Dual-core ARM Cortex-A9 (HPS)	85000	6	-
De10 Lite	https://www.terasic.com.tw/cgi-bin/page/archive.pl?Language=English&CategoryNo=218&No=1021&PartNo=2#contents	MAX 10 10M50DAF484C7G Device	Integrated dual ADCs, each ADC supports 1 dedicated analog input and 8 dual function pins	50000	4	-
De10 Standard	https://www.terasic.com.tw/cgi-bin/page/archive.pl?Language=English&CategoryNo=167&No=1081#contents	Cyclone V SX SoC—5CSX FC6D6F31C6N	Dual-Core ARM Cortex-A9 MPCore Processor	110000	6	3
De10 Nano	https://www.terasic.com.tw/cgi-bin/page/archive.pl?Language=English&CategoryNo=167&No=1046&PartNo=2#contents	Intel Cyclone V SE 5CSEBA6U23I7 device	Dual-core ARM Cortex-A9 processor	110000	6	-
De0 CV	https://www.terasic.com.tw/cgi-bin/page/archive.pl?Language=English&CategoryNo=163&No=921&PartNo=2#contents	Cyclone V 5CEBA4F23C7N Device	Nios II Processor	49000	4	-

	nts					
--	---------------------	--	--	--	--	--

Название	Объем встроенной памяти ПЛИС (Kbits)	Объем внешне й памяти ПЛИС	Объем внешней памяти HPS	Наличие коммуникац ии USB	Наличие коммуникац ии Ethernet	Наличие коммуникац ии PS/2 mouse/keybo ard	Наличие коммуникации IR Emitter/Receiver
De1 SoC	4,45	64MB (32Mx1 6) SDRAM	1GB (2x256Mx1 6) DDR3 SDRAM	USB 2.0 Host (2 порта), UART to USB	10/100/1000	1	1
De10 Lite	1,638	64MB SDRAM	0	Normal type B USB connector	0	0	0
De10 Standard	5,761	64MB (32Mx1 6) SDRAM on FPGA	1GB (2x256Mx1 6) DDR3 SDRAM on HPS	Two USB 2.0 Host Ports (ULPI Interface with USB Type A Connector) on HPS, USB to UART (Micro USB Type B Connector) on HPS	10/100/1000	1	1
De10 Nano	5570	0	1GB (2x256Mx1 6) DDR3 SDRAM on HPS	One USB 2.0 OTG (ULPI interface with USB Micro-AB connector) UART to USB (USB Mini-B connector)	10/100/1000	0	0
De0 CV	3080	64MB SDRAM, x16 bits data bus	0	Normal type B USB connector	0	1	0

Название	Наличие разъема Two 40-pin Expansion Headers	Наличие разъема One HSMC Connector (Configurable I/O Standards 1.5/1.8/2.5/3.3V)	Наличие разъема One 10-pin ADC Input Header	Наличие разъема One LTC connector (One Serial Peripheral Interface (SPI) Master ,one I2C and one GPIO interface)	Наличие разъема One 2x20 GPIO Connector (voltage levels: 3.3V)	Наличие разъема Arduino Uno R3 Connector , including six ADC channels.	Дисплей	Аудио
De1 SoC	1	0	1	1	0	0	24-bit VGA DAC	24-bit CODEC, Line-in, line-out, microphone-in jacks
De10 Lite	0	0	0	0	1	1	4-bit Resistor VGA	-
De10 Standard	1	1	1	1	0	0	24-bit VGA DAC, 128x64 Dots LCD Module with Backlight on HPS	24-bit CODEC, Line-in, Line-out, and Microphone-In Jacks
De10 Nano	1	0	1	1	0	0	HDMI TX, compatible with DVI v1.0 and HDCP v1.4	-
De0 CV	1	0	0	0	1	0	4-bit Resistor VGA	-

Название	Видеовход	Частота дискретизации АЦП	Количество каналов АЦП	Разрешение АЦП	Диапазон аналогового входа АЦП	Датчики	Питание
De1 SoC	TV Decoder (NTSC/PAL/SECAM) and TV-in connector	500Ksps	8	12 Bit	0 ~ 4.096 V	G-Sensor on HPS	12V DC
De10 Lite	-	50 MHz	1	16 Bit	-	Accelerometer	5V DC
De10 Standard	TV Decoder (NTSC/PAL/SECAM) and TV-In Connector	500Ksps	8	12 Bit	0 ~ 4.096 V	G-Sensor on HPS	12V DC
De10 Nano	-	500Ksps	8	12 Bit	0V ~ 4.096V	G-Sensor on HPS	5V DC input
De0 CV	-	-	-	-	-	-	5V DC input

Название	Кнопки	Переключатели	Светодиоды	Семисегментный дисплей
De1 SoC	6 (4 FPGA; 2 HPS Reset Buttons (HPS_RST_n and HPS_WARM_RST_n))	10	11 (FPGA x10 ; HPS x 1)	6
De10 Lite	2	10	10	6
De10 Standard	7 (5 User Keys (FPGA x4, HPS x1); 2 HPS Reset Buttons (HPS_RST_n and HPS_WARM_RST_n) 7-Segment Display x6)	10 User Switches (FPGA x10)	11 User LEDs (FPGA x10 ; HPS x 1)	-
De10 Nano	3 user Keys (FPGA x2, HPS x1); 2 HPS reset buttons (HPS_RESET_n and HPS_WARM_RST_n)	4 user switches (FPGA x4)	9 user LEDs (FPGA x8, HPS x 1)	-
De0 CV	4 Debounced Push Buttons; 1 CPU reset Push Buttons	10 Slide Switches	10 LEDs	6

Приложение 2. Листинг программы

```
import pandas as pd

import ipywidgets as widgets

from IPython.display import display

# Чтение xlsx файла в DataFrame

file_path = 'data.xlsx' # Убедитесь, что файл загружен в рабочую директорию

df = pd.read_excel(file_path)

characteristics = df.columns.difference(['Название', 'Ссылка', 'Тип ПЛИС'])

# Преобразование значений в столбцах, начинающихся с "Наличие"

for col in characteristics:

    if col.startswith('Наличие'):

        df[col] = df[col].replace({1: 'да', 0: 'нет'})

# Заменить значения "-", "nan" и другие нежелательные значения на "неважно"

df = df.fillna("неважно")

df.replace("-", "неважно", inplace=True)

# Создаем виджеты для каждой характеристики

widgets_dict = {}

labels_dict = {}

for characteristic in characteristics:

    options = list(df[characteristic].unique())

    if "неважно" not in options:

        options.append("неважно")
```



```

widgets_dict[characteristic] = widgets.Dropdown(

    options=options,

    layout=widgets.Layout(width='50%')

)

labels_dict[characteristic] = widgets.Label(

    value=characteristic,

    layout=widgets.Layout(width='50%')

)

# Функция для обработки выбора пользователя и нахождения наиболее подходящей
платы

def find_best_board(b):

    scores = {board: 0 for board in df["Название"]}

    for characteristic in characteristics:

        user_choice = widgets_dict[characteristic].value

        if user_choice != "неважно":

            for index, row in df.iterrows():

                if row[characteristic] == user_choice:

                    scores[row["Название"]] += 1

    best_board = max(scores, key=scores.get)

    print(f"Наиболее подходящая плата: {best_board}")

# Кнопка для завершения выбора и вызова функции

button = widgets.Button(description="Завершить выбор")

button.on_click(find_best_board)

```

```

# Собираем все виджеты и метки в список для отображения

widgets_list = []

for characteristic in characteristics:

    widgets_list.append(widgets.HBox([labels_dict[characteristic],
widgets_dict[characteristic]]))

# Отображаем виджеты

widgets_box = widgets.VBox(widgets_list,
layout=widgets.Layout(flex_flow='column', align_items='stretch',
width='100%'))

display(widgets_box)

# Отображаем кнопку ниже всех виджетов

display(button)

```