

УДК 621.382

Поиск критериев и оптимизация выбора программируемых логических схем для начинающих разработчиков

Алиев М.И., студент

*Россия, 105005, г. Москва, МГТУ им. Н.Э. Баумана,
кафедра «Информационные системы и телекоммуникации»*

Научный руководитель: Крупна З.П., старший преподаватель

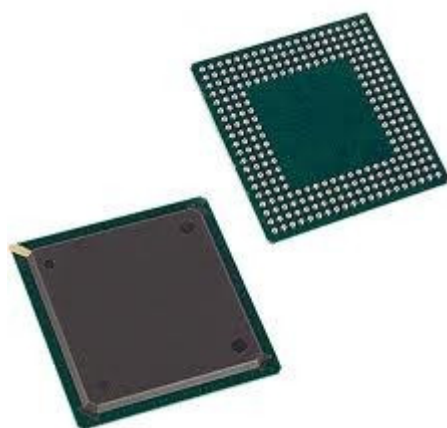
Россия, 105005, г. Москва, МГТУ им. Н.Э. Баумана

iu3@bmstu.ru

Программируемая логическая интегральная схема (ПЛИС) — один из основных видов электронных компонентов, применяемых для реализации различных цифровых устройств самой разной сложности, от двоичных счетчиков и других не сложных цифровых схем (замены стандартных цифровых интегральных микросхем низкой степени интеграции) до специализированных систем шифрования, обработки сигналов, процессоров, систем-на-кристалле и нейрочипов [1,2]. Сам термин ПЛИС появился в свое время как перевод английского названия новых (для 80-х годов XX века) устройств, PLD (Programmable Logic Devices), а затем, с увеличением логической емкости, и CPLD (Complex Programmable Logic Devices). Появившиеся в последствие принципиально отличные от CPLD устройства, получившие название FPGA (Field Programmable Gate Arrays, дословно «Вентильные (логические) матрицы, программируемые полем»), нового названия в русском языке не получили и в настоящее время, русская аббревиатура ПЛИС включает в себя и PLD и CPLD и FPGA [6]. В отличие от обычных цифровых микросхем, логика работы ПЛИС не определяется при изготовлении, а задаётся посредством программирования (проектирования). Логика работы обеспечивает встраиваемое программное обеспечение, которое может быть синтезировано из логических схем (так называемый схемный ввод), введенных в специализированных САПР ПЛИС (Altera Quartus II, Xilinx Foundation, Cadence OrCAD, Altium Designer), а также написано как на языке низкого уровня (например, AHDL), так и высокого уровня (например, Verilog, VHDL). Для написания встраиваемого программного обеспечения в последнее время активно применяются языковые конвертеры и синтезаторы (например, из C++ в Verilog или VHDL) [7]. Для записи написанного и отлаженного встраиваемого

программного обеспечения в ПЛИС используются программаторы и внутрисхемные отладочные средства, подключаемые к персональным компьютерам [2,4,5].

На сегодняшний день ПЛИС выпускаются в двух основных типах корпусов: QFP и BGA, представленных на рис. 1. Каждый из этих двух типов имеет несколько вариаций, отличающихся шагом между выводами и их размером, а также толщиной корпуса. Главное отличие между ними заключается в том, что выводы у корпусов типа QFP расположены по периметру (четырёхстороннее расположение выводов), а у корпусов типа BGA – под корпусом (нижнее расположение выводов). Эта особенность накладывает свой отпечаток на применение ПЛИС: микросхемы в корпусах QFP обладают большими размерами, чем их аналоги в корпусах BGA, но при этом их пайка не требует специального оборудования, а контроль качества пайки может быть произведен при помощи обычного увеличительного стекла, тогда как проконтролировать качество пайки BGA можно только с использованием рентгеновских лучей. Однако, корпуса BGA становятся предпочтительнее, когда проект требует большого количества пользовательских линий ввода-вывода, а также ячеек памяти функциональных преобразователей и ОЗУ, поскольку нижнее расположение выводов позволяет достичь большей плотности расположения выводов и использовать полупроводниковый кристалл большего размера.



а

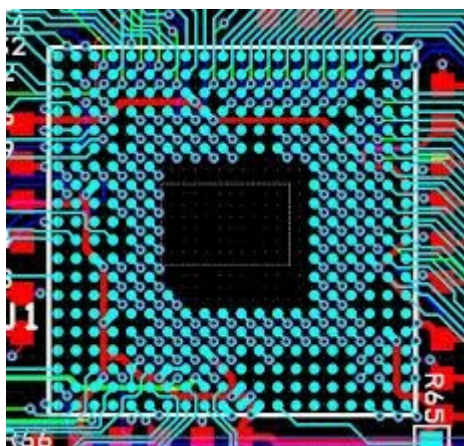


б

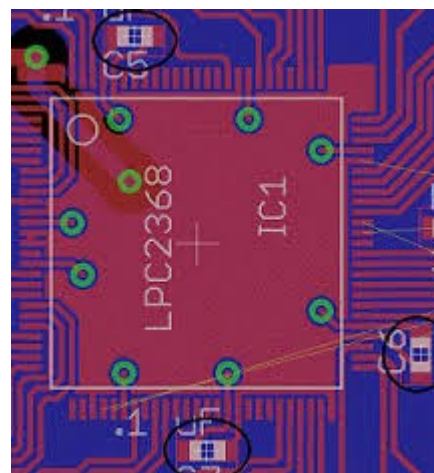
Рис. 1 Корпуса ПЛИС (а – BGA, б – QFP)

Трассировка печатной платы, на которую планируется устанавливать микросхему в корпусе BGA, является довольно сложной задачей: высокая плотность выводов приводит к необходимости использования большого количества слоев металлизации, что в свою очередь приводит к удорожанию конечного изделия. Пример трассировки фрагмента платы с микросхемой в корпусе типа BGA приведен на рис. 2а. Печатная плата, на которую планируется установить микросхему в корпусе QFP, как показано на рис. 2б,

может содержать всего 2 слоя металлизации, а необходимость увеличения их количества диктуется, в основном, требованиями по электромагнитной совместимости, согласованием импеданса и обеспечением помехоустойчивости, а не по плотности трассировки. Таким образом, получается, что корпуса типа QFP предпочтительнее для макетирования и применения в проектах студентов и начинающих разработчиков, если им не требуется большого количества линий ввода вывода или других ресурсов.



а



б

Рис. 2 Пример трассировки фрагмента печатной платы с ПЛИС (а – BGA, б – QFP)

На рынке имеется около десятка широко известных производителей ПЛИС. Лидерами являются американские компании Altera и Xilinx. Помимо них следует отметить компании Cypress Semiconductors и Lattice Semiconductor Corporation. Также нельзя обойти стороной компанию Actel, но поскольку она производит специализированные продукты, предназначенные преимущественно для военных и космических нужд, ее продукцию в данной статье рассматривать не будем. Каждая из указанных компаний производит довольно обширную номенклатуру ПЛИС, которые различаются между собой как по технологическим параметрам (технология, лежащая в основе кристалла и, как следствие, напряжение питания ядра и портов ввода-вывода, максимальная рабочая частота, новизна семейства), так и по потребительским характеристикам (логическая емкость, количество доступных пользователю линий ввода-вывода, объем внутреннего ОЗУ, стоимость). Выбрать подходящую ПЛИС для проекта начинающему разработчику, а тем более студенту, трудно. Другое дело опытные разработчики, или целые рабочие группы внутри компаний, реализовавшие не один десяток проектов. Как правило, в этом случае одним из основных критериев выбора будет наличие опыта работы с продукцией конкретной компании-производителя или даже с конкретным семейством, а также наличие профессиональной среды разработки (как правило, довольно дорогой).

Проанализировав имеющиеся на рынке близкие по характеристикам ПЛИС, приведенные в таблице, попытаемся помочь студентам и начинающим разработчикам.

Производитель	Название	Кол-во линий ввода/вывода	Цена	Кол-во логических элементов	Объем ОЗУ	Тип корпуса
Xilinx	XC3S50A-4FTG256C	144	356,36	1584	55296	LBGA 256
Lattice	iCE40HX8K-CB132	95	383,89	7680	128000	CBGA 132
Altera	EP4CE6E22C8N	91	482,6	6272	270000	QFP-144
Lattice	LFE2-6E-5FN256C	190	575,82	6000	67000	FPBGA 256
Xilinx	XC6SLX9-3TQG144C	102	631,97	9152	589824	LQFP 144
Lattice	LFXP2-5E-5QN208C	146	634,17	5000	176000	PQFP 208
Lattice	LCMXO2-4000HC-4BG332C	274	729,23	4320	222000	caBGA 332
Altera	EP4CE6F17C7N	179	754,55	6272	270000	FBGA 256
Xilinx	XC3S400A-4FGG320I	251	871,26	8064	368640	BGA 320
Xilinx	XC6SLX16-2FTG256C	186	890,71	14579	589824	LBGA 256
Xilinx	XC3S400A-4FGG400C	311	890,71	8064	368640	BGA 400
Lattice	LFXP6C-3TN144C	100	933,281	6000	95000	TQFP 144
Altera	EP3C16Q240C8N	160	1068	15408	516096	PQFP 240
Lattice	LFECP10E-3Q208C	147	1092,93	12000	317000	PQFP 208
Lattice	LFE2-12E-5FN256C	193	1151,28	12000	245000	FPBGA 256
Xilinx	XC3S500E-4PQG208I	158	1164,12	10476	368640	PQFP 208
Altera	EP4CE15E22C7N	81	1168,53	15408	504000	QFP 144
Xilinx	XC6SLX25-2FTG256C	186	1256,96	24051	958464	LBGA 256
Altera	EP4CE15F23C8N	343	1402,31	15408	504000	FBGA 484
Lattice	LFXP2-17E-5QN208C	146	1450,38	17000	311000	PQFP 208
Altera	EP4CE15F17C7N	165	1461,03	15408	504000	FBGA 256
Xilinx	XC3S700AN-4FGG484C	372	1510,57	13248	368640	BBGA 484
Lattice	LFEC15E-3F256C	195	1519,01	15400	411000	FPBGA 256
Altera	EP4CGX15BF14I7N	72	1524,52	14400	540000	FBGA 169
Altera	EP4CE10F17I7N	179	1632,05	10320	414000	FBGA 256
Lattice	LFXP2-17E-6FN484C	358	1923,08	17000	311000	FPBGA 484
Altera	EP4CGX30CF19C8N	150	2931,6	29440	1080000	FBGA 324
Altera	EP4CE40F23C7N	328	3351,81	39600	1134000	FBGA 484
Altera	EP4CGX50CF23C8N	290	5222,78	49888	2502000	FBGA 484
Altera	EP3C40Q240C8	128	6218,82	39600	1161216	PQFP 240
Altera	EP4CE55F23C7N	324	5827,59	55856	2340000	FBGA 484

Для оценки ПЛИС воспользуемся формулой:

$$R = \text{SUM}(K_i * C_i)$$

, где R – взвешенная сумма критериев C_i и K_i – весовой коэффициент каждого критерия:

- C_1 – количество входов/выходов (K_B), $K_B \leq 100$, $K=1$; $100 < K_B \leq 200$, $K=2$; $200 < K_B \leq 300$, $K=3$; $300 < K_B \leq 400$, $K=4$; $400 \leq K_B < 500$, $K=5$; $K_B \geq 500$, $K=6$.

- C_2 – цена, $C \leq 300$ руб., $K=5$; $C \leq 600$ руб., $K=4$; $C \leq 900$ руб., $K=3$; $C \leq 1200$ руб., $K=2$; $C > 1200$ руб., $K=1$;
- C_3 – количество эквивалентных вентилях (ЭВ), $6000 \leq \text{ЭВ} < 15000$, $K=1$; от 15 до 29 тыс., $K=2$; от 29 до 50 тыс., $K=3$; от 50 до 75 тыс., $K=4$; от 75 до 115 тыс., $K=5$; от 120 тыс., $K=6$;
- C_4 – объем ОЗУ, $\text{ОЗУ} \leq 517\text{Kbits}$, $K=1$; $\text{ОЗУ} \leq 517\text{Kbits}$, $K \leq 2$; $\text{ОЗУ} \leq 756\text{Kbits}$, $K=3$; $\text{ОЗУ} \leq 2\text{Mbits}$, $K=4$; $\text{ОЗУ} \leq 4\text{Mbits}$, $K=5$; $\text{ОЗУ} > 5\text{Mbits}$, $K=6$;
- C_5 – относительная стоимость монтажа (корпус типа BGA $K=0$; корпус типа QFP $K=1$)

Графики зависимости взвешенной суммы и основных параметров ПЛИС от стоимости представлены на рис.3. Из графиков видно, что в основном рост цены сопровождается увеличением объема доступных ресурсов, что, в свою очередь, приводит к росту значения R . Также видно, что резкий рост одного из параметров не приводит к существенному увеличению значения R и, напротив, одновременный рост нескольких параметров приводит к резкому росту взвешенной суммы.

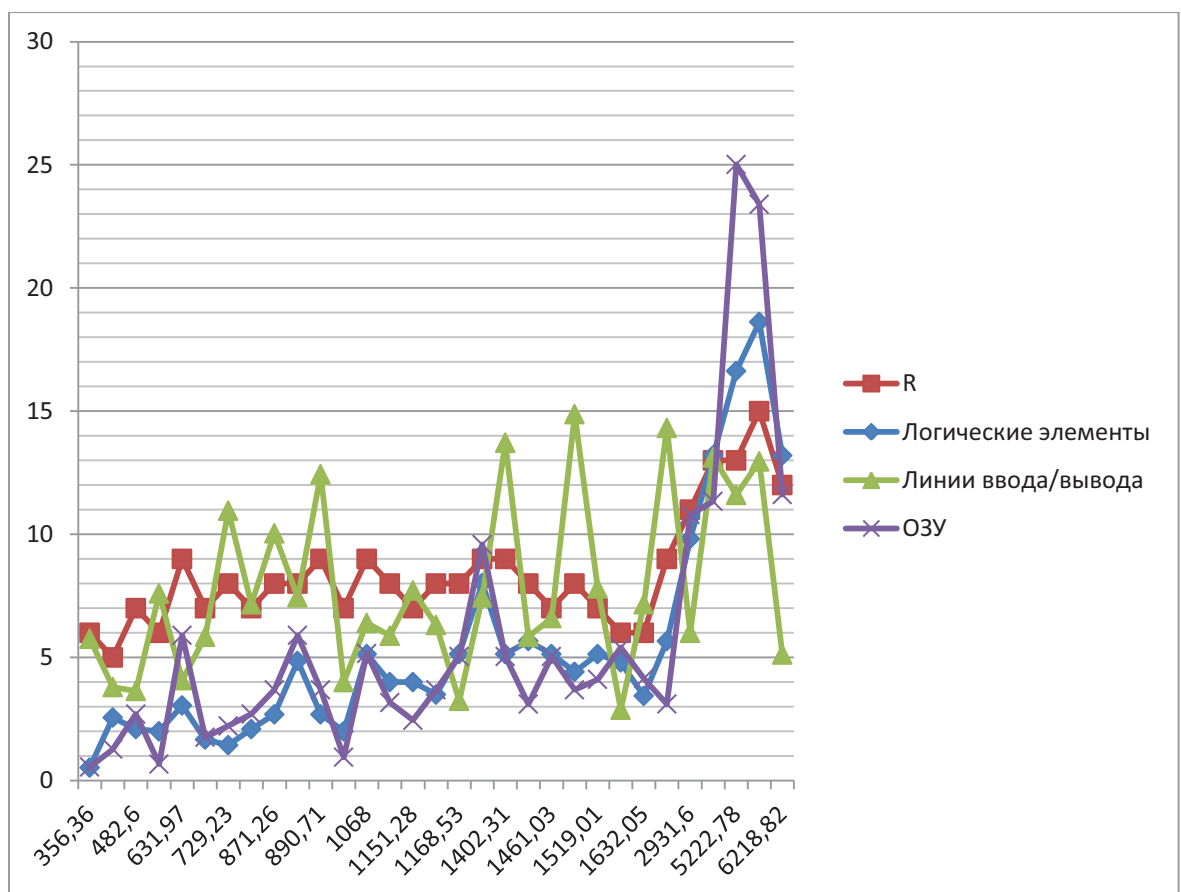


Рис. 3. Зависимость значения R и параметров ПЛИС от цены

Список литературы

1. Круппа З.П. Аппаратная реализация алгоритма криптографического преобразования в соответствии с ГОСТ 28147-89 на базе ПЛИС фирмы Altera // Безопасность информационных технологий. 2003. №1. С. 36-40.
2. Угрюмов Е. П. Программируемые компоненты устройств и систем на кристалле: Учеб. СПб: Изд-во СПбГЭТУ «ЛЭТИ», 2013. 120 с.
3. Бибило П.Н. Основы языка VHDL: Учебное пособие. Изд. 5-е. М.: Книжный дом «ЛИБРОКОМ», 2012. 328 с.
4. Федоров С.В. Основы языка Verilog. М.: МГТУ им. Н.Э. Баумана (каф. ИУ-3) 2009. 70 с.
5. Угрюмов Е.П. Цифровая схемотехника: учеб. пособие для вузов. 3-е изд., перераб. и доп. СПб.: БХВ-Петербург, 2010. 816 с.
6. Свободная интернет-энциклопедия. Режим доступа: www.ru.wikipedia.org (дата обращения 15.02.2014).
7. Официальный сайт Inferno: C++ to Verilog Converter. Режим доступа: <http://inferno-cpp2v.sourceforge.net/> (дата обращения 15.02.2014).