# Езици за описание на апаратната част при СПЛ – VHDL, Verilog, SystemC. Програмируеми интегрални схеми – видове, проектиране и програмиране.

## 1. Езици за описание на апаратната част при СПЛ.

В предишна лекция бе споменато, че най-често използваните езици за описание на апаратната част са Verilog, VHDL и Системен С (SystemC). В тази лекция, ще бъде представена допълнителна информация за тези езици.

Verilog е език за описание на апаратната част, чрез който се изграждат електронни устройства и системи. Verilog се използва за проектиране, верификация (проверка) и реализация на ЦУС. Има възможност за верификация също така и на аналогови и смесени (цифрови и аналогови) системи.

Езиците за описание на апаратната част се отличават от езиците за описание на програмната част по няколко основни характеристики. Тук се изгражда концепцията на конкуренция – паралелно изпълнение на твърденията, както и съществуват зависимости между сигналите.

Притежава два оператора за присвояване – блоков (=) и не блоков (<=). Не блоковото присвояване позволява на инженерите проектанти на ЦУС да описват машини на състоянията, без да е необходимо да декларират временни променливи за съхранение. Това е една от основните характеристики на езика. В самото начало, когато започва използването на Verilog, се установява огромно подобрение на производителността на инженерите проектанти. До този момент те са използвали графични модели и специални CAD програмни пакети за симулиране на електронни компоненти. Създателите на езика Verilog са имали за цел да изработят език, който да е със синтаксис, подобен на С и С++, тъй като до тогава това са били основните езици, които са били използвани за описание на програмната част на микропроцесорните системи.

Verilog различава малки от големи букви при описанието на ЦУС. Съдържа ключови думи, които се използват за контролиране на процесите (if/else, for, while, case) и контрол на операторите. Основните синтактични различия са при деклариране на променливите и разграничаване на процедурните блокове – използват се ключовите думи begin и end.

Основният механизъм, който се използва е йерархията на отделните модули. Модулите участват в йерархията на проекта и комуникират с другите модули чрез набор от декларирани входове, изходи и двупосочни такива. Като цяло, модулите могат да са комбинация от декларации, конкурентни и последователни блокове и подмодули. Последователните елементи са разположени в блок, ограничен от ключовите думи begin и end. Изпълняват се последователно вътре в блока. Но сами по себе си, тези блокове се изпълняват конкурентно, което характеризира Verilog като език за описание на апаратната част. Използват се четири състояния на сигналите – логическа нула, логическа единица, високо импедансно и не дефинирано. Използва се и т.нар. сила на сигналите – силни, слаби и т.н. Това позволява абстрактно моделиране на споделените сигнали, където много изчислителни ресурси използват обща шина. По този начин конкретната логическа стойност се определя като се използва зададената силата на сигналите и техния източник.

В езика Verilog множествата от условия са синтезируеми (превръщат се в конкретни електронни схеми). Това се изпълнява чрез частта от CAD програмния пакет наречена синтезираща програма (модула за синтез). Тя преобразува направеното описание чрез езика Verilog в логически еквивалент, който се състои само от прости

(основни) логически елементи и модули. Кратка история на езика Verilog е представена на следващите редове.

Първоначалното му развитие започва през 1983/84 година, като първия съвременен език за описание на апаратна част. Негови автори са Phil Moorby and Prabhu Goel. След това се появяват различни модификации (версии) на езика като:

- Verilog 95 – развива се като еквивалент на VHDL, който да позволява стандартизация. След това езика Verilog се превръща в стандарт IEEE – 1364-1995.
- Verilog 2001 – подобрение на Verilog 95, което цели да се изключат недостатъците, които потребителите са открили в предишната версия. Това също се превръща в стандарт IEEE – 1364-2001. Счита се, че това е едно от най-значителното подобрение на езика. Преди това инженерите проектанти е трябвало да изпълняват операциите на битово ниво (например, преносът при осем битово събиране). Същата тази функционалност се реализира много по-сбито във версията на езика Verilog 2001, като се използват вградените оператори: за събиране (+), за изваждане ( - ), за деление (/), за умножение (*) и др. Входът (изходът) е подобрен чрез някои допълнителни системни задачи. Също така, са добавени и няколко допълнителни синтактични подобрения, които целят подобряване на четимостта на направеното описание – конструкцията always @, имена на параметри, използване на програмиране в стила на езика C. Verilog 2001 е по-високата версия на Verilog, която се поддържа от повечето EDA софтуерни пакети, които се предлагат на пазара.
- Verilog 2005 – тук се използва стандарт IEEE 1364-2005. Той съдържа някои незначителни корекции, спецификации и някои допълнителни характеристики като например, ключовата дума uwire.
- Verilog-AMS – отделна част на стандарта за Verilog, която интегрира аналоговите и смесените сигнали, като ги моделира с традиционния език за описание на апаратната част Verilog.
- SystemVerilog – представлява най-сериозното подобрение на Verilog 2005. Притежава много нови характеристики и възможности. Тук са добавени възможностите за проектиране-верификация и проектиране-моделиране.

Друг език за описание на апаратната част е VHDL. Кратка история на създаването на VHDL е представена на следващите редове.

Първоначално се разработва по заявявка от департамента занимаващ се с проблемите на отбраната на САЩ (US Department of Defense), с цел да се опише поведението на специализираните приложения (ЦУС), които са били използвани дотогава т.е. VHDL е създаден като алтернатива на сложните и съдържащи някои неточности описания, които са били налични до този момент.

Разработени са също така и логически симулатори, които да работят с VHDL файловете. Следващата стъпка е била да се разработят инструменти за логическия синтез на описанията на VHDL. Те могат да отделят RAM памет, броячи, аритметични блокове и др. и да ги реализират според желанието на инженера проектант. Това означава, че създаденото описание на VHDL е трябвало да може да се синтезира при различни условия, изисквания, скорост и електрическо захранване.

VHDL е подобен на програмния език Ada – като концепция и синтаксис. VHDL има създадена специална конструкция, която да манипулира паралелизма при проектирането на ЦУС, което го отличава от Ada. Подобно на нея, VHDL не различава малки от големи букви при създаденото описание. Много от останалите характеристики на VHDL не са налични в Ada като например използваните Булеви оператори. VHDL позволява масивите да бъдат индексирани в двете посоки (нисходящо и възходящо), докато езиците за описание на програмната част обикновено позволяват само възходящото индексиране. Причината за толкова голямото сходство между двата езика

е, че департамента занимаващ се с проблемите на отбраната на САЩ изрично е изискал това да бъде така, с цел да се избегне повторното откриване и тестване на вече добре познатите концепции налични в програмния език Ada.

- Първоначалната версия на езика става стандарт IEEE 1076-1987. Той включва голям набор от типове данни (целочислени и реални), логически (битови и Булеви), символи и времеви, масиви, стрингове и т.н. При тази версия на езика остава един неразрешен проблем. Това са многото възможни варианти на логическите нива – без ниво, силен, слаб, неопределен. Това налага да се създаде и използва стандарта (програмния пакет) IEEE 1164. Той дефинира 9 логически типа за даден сигнал.
- Следващото подобрение е стандарта IEEE 1076-1993. Езикът става по-последователен, дава по-големи възможности при именуването. Също така, позволява да се използва стандарта ISO-8859-1 за отпечатване на символи, добавя се и оператор xnor.
- Някой незначителни промени стават при стандартите 2000 и 2002 – езика за описание на апаратната част VHDL придобива характерни за C++ черти.
- Следват редица подобрения наа стандарта IEEE 1164. Те имат за цел да дообогатят функционалността на езика. При стандарт IEEE 1076.2 се подобрява контрола върху реалните и комплексни типове данни. Стандартът IEEE 1076.3 въвежда използването на знакови и беззнакови аритметични оператори върху вектори.
- Стандартът IEEE 1076.1 е популярен като VHDL-AMS. Той се занимава с аналогови и смесени сигнали.
- През 2006 година, техническият комитет на Accellera разработи новата версия на езика VHDL 2006. От една страна, това е версия, която е напълно съвместима с предишните, но от друга страна това става стандарт, които прави употребата и манипулирането на описанието направено на VHDL много по-лесни. Тук подстандартите 1164, 1076.2, 1076.3 се обединяват в стандарт 1076, като се добавят много нови оператори и функции. Това изключително много подобрява синтезирането на VHDL описанието, прави файловете, които служат за проверка и верификация на ЦУС (testbenches) още по-приложими. Като цяло, позволява по-широка и по-лесна употреба на езика.
- През 2008 година, отново Accellera подобряват езика и се стига до VHDL 4.0. Той се превръща и в стандарт IEEE 1076-2008.

VHDL принадлежи към езиците за описание на апаратната част с общо предназначение. Той изисква използването на специална програма (симулатор) за проверка на работоспособността на ЦУС. Налични са разнообразни симулатори, които генерират изпълним код. Симулаторът чете и записва файлове на компютъра.

Тъй като е език с общо предназначение, това позволява HDL да се използва и за да се създават файлове, които се използва за верификацията на функционалността на ЦУС. VHDL е език с много специализирана терминология. Неопитен специалист би могъл относително лесно да направи описание на ЦУС, което успешно да бъде симулирано, но не може да да бъде успешно синтезирано (превърнато) в реално устройство или система.

Системен С също принадлежи към групата на езиците за описание на апаратната част. Той най-вече се възприема като език за описание на системи, тъй като се използва за проектиране на електронни схеми, моделиране на нива, поведенческо моделиране и синтезиране на електронната система на високо ниво.

Системен С представлява набор от библиотеки и макроси налични в езика за описание на програмната част C++. Това прави възможно симулирането на конкурентни процеси, всеки от които е описан съобразно правилата на C++. Обектите, описани по този начин могат да комуникират в симулирана среда за работа в реално

време. Могат да се използват готови библиотеки, или потребителят може сам да създаде свои.

Поведенческите процеси могат да се използват произволен брой пъти. Заделят се ресурси и за йерархично определените процеси.

Възможно е да се каже, че Системен С (SystemC) притежава семантични прилики с VHDL и Verilog, но като цяло той е език от по-високо ниво. Той дава възможност да се използват обектно-ориентирани модели и базови класове. Нещо повече, Системен С е едновременно език за описание и симулационно ядро. Направеното описание (написаният код) се компилира едновременно с библиотечната симулация и се получава изпълним модел, който притежава поведенческите черти на описания модел. Кратка история на създаването на езика Системен С е представена на следващите редове:

- Инициативата на Open SystemC е обявена през 1999 г.
- SystemC V0.91 и SystemC V1.0 са създадени през 2000 г.
- SystemC V2.0 спецификация и V1.2 Beta source code са създадени през следващата година.
- SystemC 2.0.1 LRM (Language Reference Manual, ръководство за описание на езика) е създадено през 2003 г.
- SystemC 2.1 LRM and TLM 1.0 (Transaction-Level Modeling, стандарт за моделиране на нивата) е създаден през 2005 г. През същата година е създаден и стандарта IEEE 1666-2005.
- SystemC v2.2 е създаден през 2007 г.
- TLM 2.0 е създаден през следващата година.

На редовете по-долу е представена кратка характеристики на езика Системен С:

- Модул – основен блок за изграждане на йерархията на езика. Създадените модели (описания) на Системен С обикновено са изградени от няколко различни модула, които обменят информация чрез портове. Модулите могат да бъдат считани за градивните единици на езика.
- Порт – позволява комуникацията от модула към останалите елементи, които най- често са също модули.
- Процес – главен изчислителен елемент. Тук процесите са също конкурентни.
- Канал – това представлява комуникационен елемент. Могат да бъдат проводници или сложни комуникационни механизми като например буфер, или комукационни канали. Най-простите канали са сигнал, буфер, памет от тип FIFO ( First In First Out, първи влязъл, първи излязъл), семафор, интерфейс (използват се от портовета за комуникация с каналите), събития (осъществяват синхронизацията между каналите).

Накрая, ще бъдат дадени примерни описания на логически елемент от тип ИЛИ като за това ще бъдат използвани трите споменати HDLs.

- На езика Verilog

```
module OR2 (x, y, F);
input x, y;
output F;
reg F;
always @(x or y)
begin
F <= x|y;
end
endmodule
```

- На VHDL

```
library ieee;
use ieee.std_logic_1164.all;
entity OR2 is
port (x, y : in std_logic;
      F : out std_logic);
end OR2;
architecture beh of OR2 is
begin
process (x, y)
begin
F <= x or y;
end process;
end beh;
```

- На Системен С

```
#include “systemc.h”
SC_MODULE (OR2)
{ sc_in<sc_logic> x, y;
sc_out<sc_logic> F;
SC_CTOR (OR2)
{SC_METHOD (comblogic);
Sensitive <<x <<y;
}
void comblogic ()
{F.write (x.read ()| y.read ());
}
};
```

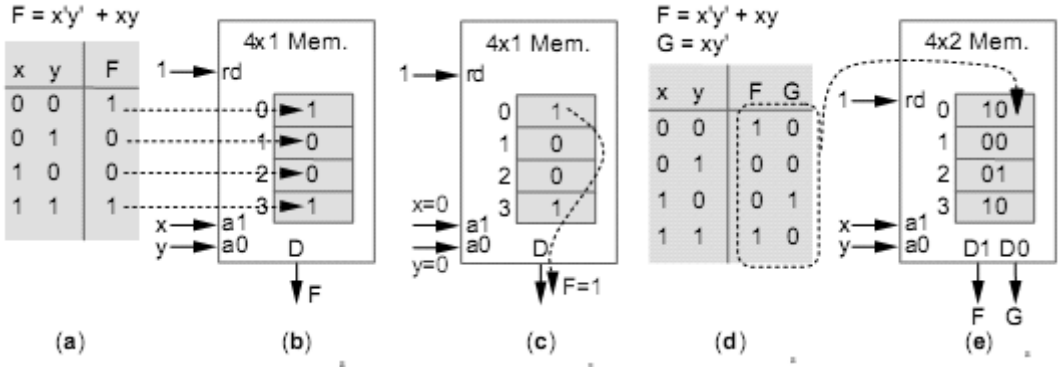
## 2. Проектиране и програмиране на FPGA и CPLD.

Програмируемите интегрални схеми са от тип CPLDs (Complex Programmable Logic Devices, сложни програмируеми логически устройства) и от тип FPGA (Field Programmable Gate Arrays, програмируеми области от логически елементи). Те осигуряват необходимото бързодействие за голяма група от приложения като тактовата честота е в обхвата между 50MHz и 400MHz. Има достигнати честоти от порядъка на 1GHz при по-новите поколения FPLDs.

Вътрешно CPLDs и FPGAs съдържат много на брой еднакви програмируеми логически елементи (LE), наречени клетки. Програмируемите логическите елементи (клетки) могат да се свържат по подходящ начин и да образуват необходимата логическа функция, като изхода може да се свърже към входа на един или два тригера. Програмируемите логически елементи могат да бъдат свързани в колона или в матрица в силициевия кристал. Освен това логическите елементи са свързани помежду си също с програмируеми елементи, което дава още по-голяма гъвкавост на проектанта при реализиране на желаната логическа функция. В добавка към това може да се каже, че много често програмируемите логически елементи съдържат програмируеми логически връзки към съседните на тях клетки, които осигуряват по-малко закъснение на сигналите. Възможно е при проектирането на дадено електронно устройство конкретната ИС от тип CPLDs и FPGAs да не може да осигури нейното реализиране

поради изчерпване на възможните програмируеми логически елементи. В този случай е необходимо да се избере CPLD или FPGA с по-голям капацитет (по-голям брой LE).

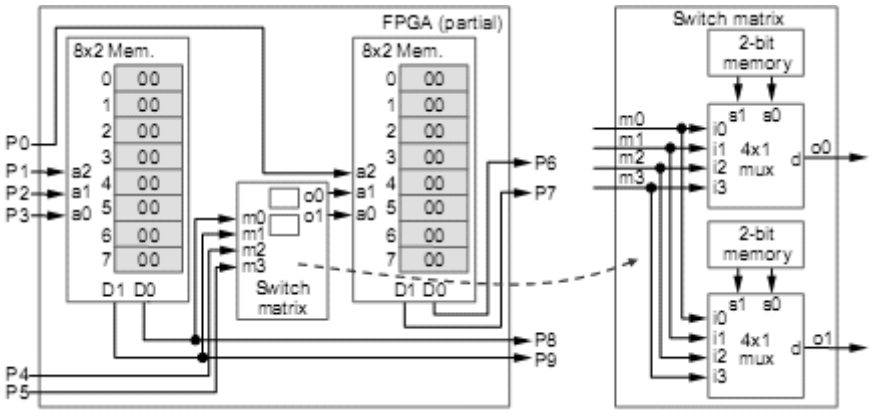
Основната идея, използвана за реализирането на FPGA е, че паметта може да се използва за реализирането на комбинационни логически схеми. На фигурата по-долу е показано как може да стане това.




Първо се започва с описание на една предварително избрана логическа функция F на две променливи x и y. Таблицата на истинност е показана най-ляво на фигурата. След това е показано реализирането на функцията като се използва 4x1 памет (4 запомнящи клетки, всяка от които запомня информация от 1 бит). Вижда се, че реализацията е възможна като като променливите са свързани с адресните линии на паметта (a0 и a1), а резултата на логическата функция се получава на шината за данни на паметта (D).

Друга логическа функция е описана със следващата таблица на истинност. Реализирана е с използването на 4x2 памет. Тук ще бъде заведено понятието lookup table. Паметта използвана при реализирането на комбинационни логически схеми се нарича lookup table (LUT).

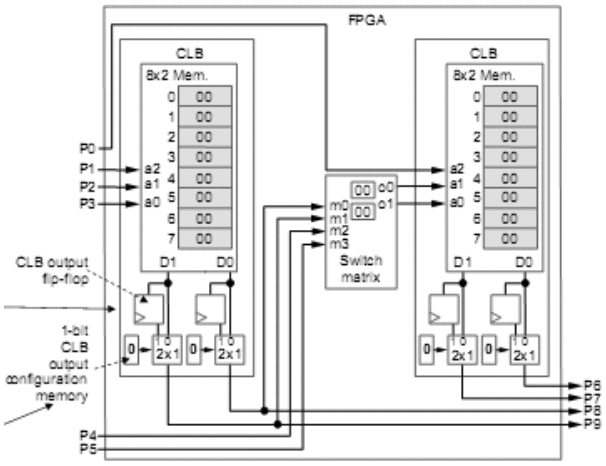
За реализирането на по-сложни комбинационни устройства е необходимо свързането на повече от една lookup table. Това се реализира чрез програмируеми връзки наречени матрични ключове (switch matrices). Това е показано на фигурата по-долу.



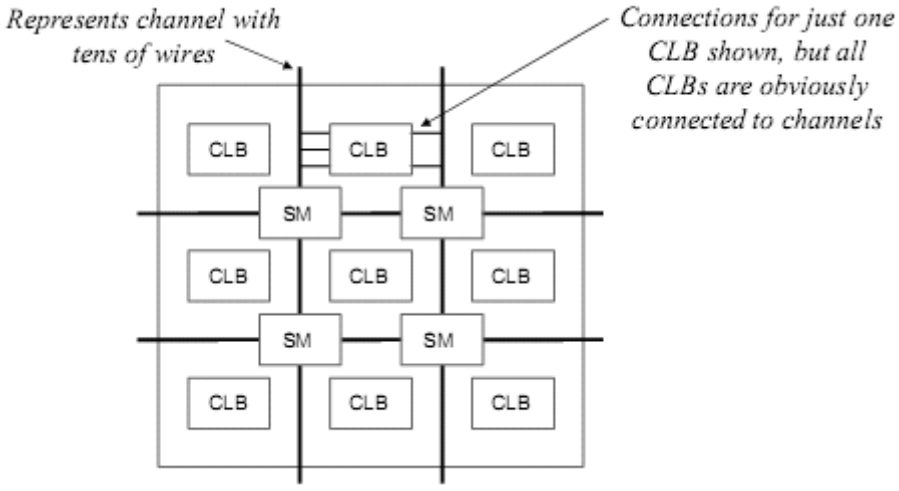

От фигурата се вижда, че проводниците за данни D0 и D1 (реализираните логически функции)са свързани посредством мултиплексор към адресните сигнали (логическите променливи) на следващата lookup table. Селектиращите (управляващите) входове на мултиплексора се командват от памет. По този начин, чрез запис в тази памет (програмно) може да бъде зададено кой от сигналите за данни да бъде свързан

към адресния сигнал на следващата памет. Това дава възможност за реализиране на различни по сложност и разрядност комбинационни логически устройства.

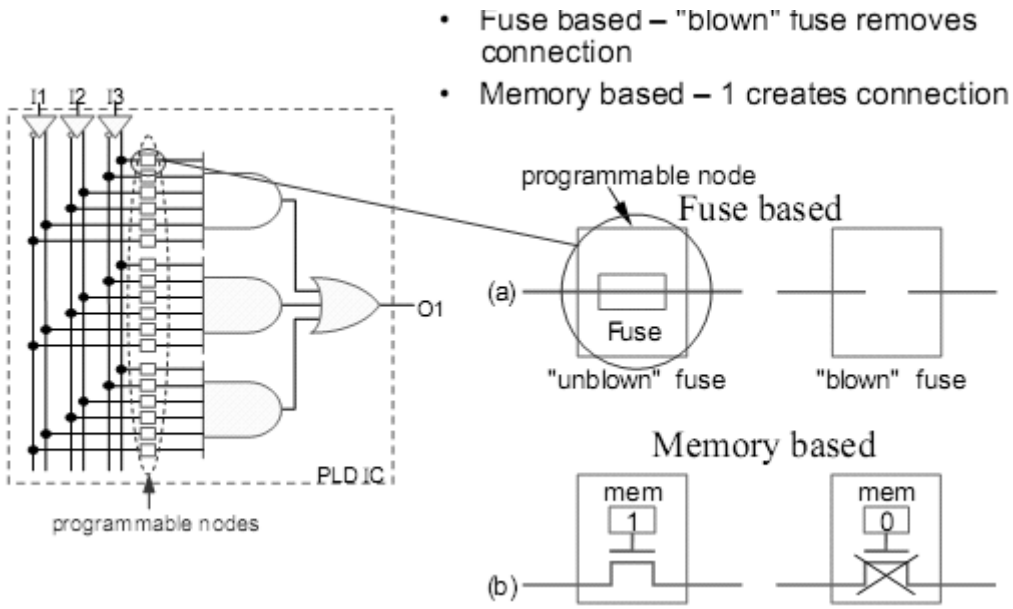
При програмирането на FPGA се записват нули и единици както в lookup tables, така и в паметта отговаряща за свързването им. Програмират се също така и връзките на сигналите с изводите на използваната FPGA. Що се отнася до реализирането на последователностните схеми (елементите с памет) те са реализирани предварително и чрез програмирането на връзките те могат да се използват или не в даден проект. Това е показано на следващата фигура.




Съвкупността от lookup table и даден тригер се нарича конфигуруем логически блок (Configirable Logic Block, CLB). Дадена FPGA може да се представи като съвкупност от CLB и програмируеми матрични ключове (Switch Matrices, SM). Това е показано на фигурата по-долу.




Другият тип програмируема ИС е CPLD. Представлява съвкупност от предварително създадени “И” и “ИЛИ” логически елементи. При тях могат да се програмират връзките между отделните логически елементи, като това е показано на фигурата по-долу.




Разликата между CPLDs и FPGAs е че, CPLDs предлагат по-предсказуеми и по-високи времеви параметри (честоти), докато FPGAs предлагат по-голяма интегрална плътност. При големите по обем програмируеми интегрални схеми се използват специални свързващи линии, които осигуряват по-малко закъснение на сигнала. Тези свързващи линии са свързани към вътрешна бързодействаща шина. Тази шина се използва за да разпредели тактовия сигнал до всички тригери в конкретната ИС, като същевременно целта е да се намали закъснението на тактовия сигнал. Ако не се използват специални линии за тактовия сигнал, той може да се разпространява като обикновен сигнал. При това положение обаче, вследствие на различните закъснения, той пристига по различно време до различните тригери, което може да предизвика нестабилност на работата на създаденото електронно устройство. Честа практика е при изграждането на сложни електронни устройства да се използва специална линия, по която да се разпространява тактовия сигнал.

Фирмите – производители на програмируеми интегрални схеми са Xilinx, Altera, Actel, Atmel и други, като първите две фирми осигуряват 80% от производството и продажбата на програмируеми интегрални схеми. В тази връзка, ще се спра на произвежданите CPLD и FPGA от тях.

**3.Видове фамилии FPGA и CPLD произвеждани от фирмите Xilinx и Altera.**

Видове фамилии FPGA и CPLD произвеждани от фирмата Xilinx:

- Virtex®-7 – Оптимизирана за получаване на най-голямо бързодействие и FPGA с най-голям капацитет. Реализирано е 2X подобрение в бързодействието. Интегралната схема предлагаща най-големи възможности на базата на SSI (Stacked Silicon Interconnect) технологията.

- Virtex®-6 – Тук се използва ASMBL™ (Advanced Silicon Modular Block, Авангарден Силициев Модулен блок) от трето поколение. В самата фамилия могат да се разграничат и подфамилиии - LXT, SXT, и HXT. Фамилията е изградена на базата на 40nm интегрална технология като са използвани медни свързващи проводници. Virtex®-6 позволява да се изграждат много сложни и бързодействащи електронни системи. Що се отнася до подфамилиите то можем да споменем следните неща. Virtex®-6 LXT – предназначена е за изграждане на сложни логически системи, като се предлага и изпращане и получаване на цифровата информация по последователен начин. SXT – служи за цифрова обработка на сигналите и също се предлага и

изпращане и получаване на цифровата информация по последователен начин. HXT осигурява най-широка честотна лента на предваната цифрова информация като последната се изпраща и получава последователно.

- Spartan®-6 – осигурява ниска цена на едросерийното производство на електронни изделия. Фамилията има тринадесет членове имащи от 3,840 to 147,443 програмируеми логически клетки (logic cells). Консумираната мощност от интегралните схеми от тази фамилия е намалена на половина в сравнение с предишните разновидности. За изграждането й се използва двойнорегистрова 6 входна LUT. Подобрена е защитата на реализираните IP, като се използва криптиращия алгоритъм AES и защита от типа DNA (device DNA protection). Съществуват две подфамилии означени с буквите - Spartan-6 LX FPGA оптимизирана за изграждането на логически устройства и LXT – осигурява бързо предаване и получаване на цифровата информация по последователен начин.

- EasyPath™-7 FPGAs – Осигурява най-ниска цена като същевременно се постига виско бързодействие Може да се използват при разработката FPGA от тип Kintex™-7 и Virtex®-7, за да се използва тяхната програмируемост. След като проектът стигне до етап, че няма нужда да се програмируемостта на горе посочените ИС, може да се премине към EasyPath-7 FPGAs за да се намали цената. Преходът към EasyPath-7 FPGAs може да се извърши за 6 седмици и не изисква промяна в проектираното устройство или промяна в използваната FPGA.

- EasyPath™-6 FPGAs – При нея за проектирането може да се използва Virtex®-6, като след това бъде изработено по-евтина интегрална схема от фирмата Xilinx за период от 6 седмици. По-този начин се съчетават по-голямото бързодействие с по-ниската цена на ИС.

- Xilinx CoolRunner™-II CPLDs – осигурява високо бързодействие и леснота на използването й. При разработването на електронни устройства може да се използва безплатния продукт Xilinx ISE® WebPACK.

Освен гореизброените фамилии фирмата Xilinx произвежда и FPGA от тип Kintex-7 и Artix-7, както и подобрени FPGA от тип Kintex UltraScale и Virtex UltraScale.

Видове фамилии FPGA и CPLD произвеждани от фирмата Altera:

- Stratix ® 10 FPGAs – Произведена е по технология на фирмата Intel, 14 nm 3D Tri-Gate. Предлага 2x увеличение на бързодействието. 70% по-малко консумация на електрическа мощност в сравнение с предишното поколение FPGA. Треото поколение ИС с влизащ в състава и микропроцесор. Хетерогенна многомодулна 3D архитектура вл;ючваща в състава си SRAM, DRAM и ASICs.

- Stratix ® V FPGAs – Най-високо бързодействие, най-висока степен на интеграция като е използван 28 nm технологичесн процес, ИС предлагаща голяма гъвкавост на предложените решения. Вградени 28 Gbps и 12.5 Gbps приемопредаватели, вградени IP блокове и лесна за потребителя частична реконфигурация. 30 процента по-малка консумирана мощност в сравнение със Stratix® IV FPGAs;

- Stratix ® IV FPGAs - Програмируемите ИС от тази фамилия високо-бързодействие на реализираните електронни системи, които могат да бъдат с висока степен на сложност. Има представени на пазара следните FPGA серии – Stratix (2002), Stratix GX (2003), Stratix II (2004), Stratix II GX (2005), Stratix III (2006) и Stratix IV (2008). Stratix FPGAs имат ASIC еквивалентна ИС. По този начин се съчетават изделия с ниска цена и нискорисковото проектиране като се използва FPGA. Stratix IV FPGAs осигуряват най-високата интегрална плътност, най-високо бързодействие и най-ниска консумирана мощност сред всички FPGAs изработени по 40nm технологичен процес.

Съществуват и различни разновидности на тези интегрални схеми E – с разширени възможности и GX и GT съдържащи приемо- предаватели.

- Stratix III FPGAs са най-бързодействащите и с най-ниската консумация на електрическа мощност FPGA ИС изработени по 65 nm интегрален технологичен процес. Съществуват разновидности L – за изграждането на логически цифрови системи и E – предназначени за цифрова обработка на сигналите.

- Stratix II и Stratix GX – за изграждането им се използва адаптивна архитектура на логическия модул, използващ по-бързодействащата 8 входова LUT.

- Stratix и Stratix GX са първите членове на произвежданата от Altera FPGA от тип Stratix. Чрез тях могат да изграждат бързодействащи електронни системи.

- Cyclone ® FPGA серии от програмируеми ИС. Дават възможност да се реализират устройства с ниска консумация на електрическа мощност съчетано с ниска цена на използваната ИС. Има 4 поколения от тези серии – Cyclone (2002), Cyclone II (2004), Cyclone III (2007), Cyclone IV (2009) и Cyclone V (2011).

- Cyclone® V FPGAs – Осигурява най-ниска цена и най-ниска консумация на електрическа мощност, съчетани с бързодействие, което предполага използването й в едросерийно електронно производство. 40 процента по-малка консумирана мощност в сравнение с предишното поколение FPGA, ефективна способност за логическа интеграция, интегрирани приемопредаватели, както SoC FPGA варианти включващи в състава си ARM® базирана микропроцесорна система (HPS).

- Cyclone IV FPGAs са програмируемите ИС от тип FPGA с най-ниска цена на пазара, а така също и с най-ниска консумация на електрическа мощност. Предназначението им за вграждане в едросерийни устройства и системи, водещо до по-ниска цена.

- Cyclone III FPGAs представляват комбинация от ниска цена, висока степен на функционалност и всичко това съчетано с оптимизирана консумация. Важно е сътрудничеството с тайванска фирма благодарение на което се постига ниската консумация на електрическа мощност и енергия.

- Cyclone II осигурява високо бързодействие на разработваните модули, съчетано с ниска цена на ИС.

- Cyclone е първият член на серията разработена от фирмата Altera.

- Arria 10 FPGAs and SoCs reinvent the midrange by delivering a 15% performance gain over current high-end FPGAs and up to 40% lower power than previous midrange devices.

- Arria® V family of FPGAs offer the highest bandwidth and deliver the lowest total power for midrange applications, such as remote radio units, 10G/40G line cards, and broadcast studio equipment. There are five targeted variants, including SoC variants with a dual core ARM® CortexTM-A9 hard processor system (HPS) to best meet your performance, power, and integration needs.

- Arria® FPGAs са интегралните схеми предназначени за реализирането на устройства и системи с ниска консумация на електрическа мощност и енергия съчетано с ниска цена на ИС и възможността да изпраща и получава информация последователно във времето. Състои се от серийте Arria GX (2007) и Arria II GX (2009).

- Arria GX FPGAs е ИС с най-ниска консумация на електрическа енергия осигуряваща предването на последователната информация със скорост 3.75 Gbps. Изработва се по 40 nm интегрален технологичен процес.

- Arria GX е оптимизирана ценова архитектура с приемо предватели достигащи скорости до 3.125 Gbps. Изработва се по 90 nm интегрален технологичен процес.

Съществуват следните разновидности на CPLDs от серията MAX – MAX 7000S (1995), MAX 3000A (2002), MAX II (2004), MAX IIZ (2007) и MAX V (2010):

- MAX V CPLDs – Предлага надеждни решения и до 50 процента по-малко консумирана мощност в сравнение с конкурентни CPLDs. Предлага също така и ниска цена посредством архитектура, която интегрира предишните функции. Бързодействаща CPLD изградена чрез енергомезависима архитектура;

- MAX II CPLDs - Осигуряват ниска консумация на електрическата мощност и енергия съчетано с ниска цена на ИС. Реализираните чрез MAX II проекти се съхраняват дори и при изключването на електрическото напрежение. Предназначението й е за реализирането логически и преносими устройства, а така също и при реализирането на мобилните телефони.

- MAX IIZ консумират нулева електрическа мощност и енергия (Zero power).

- MAX 300A са ценово оптимизирани и предназначени за вграждани в едросерийно производство на електронни устройства.

- MAX 7000S може да се използва в промишлени, военни и комуникационни устройства, работещи с ниво от 5V на техните входно – изходни изводи.