



# طراحی خودکار مدارهای دیجیتال جلسه ۱

---

روح الله دیانت

دانشگاه قم - دانشکده فنی - گروه مهندسی کامپیوتر

## منابع درس

---

❖ مرجع شماره ۱: **کتاب** Verilog HDL : A Guide to Digital Design and Synthesis .  
**نویسنده:** Samir Palnitkar . **ناشر:** Prentice Hall . **ویرایش دوم (۲۰۰۳).**

❖ مرجع شماره ۲: **کتاب** HDL Chip Design : A Practical guide for Designing,  
Synthesizing and Simulating ASICs and FPGAs using VHDL or Verilog Rose.  
**نویسنده:** Douglas J. Smith . **ناشر:** Doone . **(۱۹۹۷)**

# نحوه ارزیابی

---

❖ تمرینات تئوری و برنامه نویسی: ۳ نمره

❖ میان ترم: ۵ نمره

❖ پایان ترم: ۱۲ نمره

❖ تمارین به ایمیل [rdianat@qom.ac.ir](mailto:rdianat@qom.ac.ir) ارسال گردند.

# سیستم‌های رقمی و مزایای طراحی رقمی

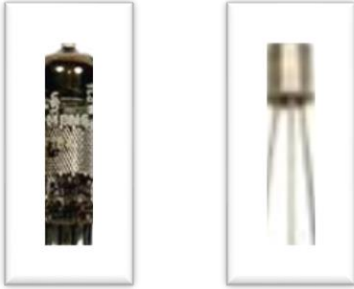
---

❖ هم اکنون، برخلاف سابق، اکثر سیستم‌ها به صورت رقمی (Digital)، طراحی می‌شوند. مانند دوربین دیجیتال (در مقابل دوربین‌های آنالوگ)

## ❖ مزایا:

- سادگی طراحی.
- مصونیت در برابر نویز (Noise immunity)
- سادگی مجتمع‌سازی (Integration)

# مراحل رشد و توسعه طراحی سیستم های دیجیتال



❖ طراحی با لامپ خلا (vacuum tubes)

❖ استفاده از ترانزیستور.

❖ استفاده از مدارهای مجتمع (Integrated circuit) یا همان Ic ها.

◦ SSI (Small Scale Integration): ده ها گیت منطقی

◦ MSI (Medium Scale Integration): صدها گیت

◦ LSI (Large Scale Integration): هزاران گیت

◦ VLSI (Very Large Scale Integration): صدها هزار گیت



# روش سنتی یک سیستم دیجیتال و مشکلات آن

- ❖ روش عادی (سنتی) طراحی یک سیستم دیجیتال شامل تعیین قطعات و قرار دادن آنها روی یک بردبرد یا PCB و اتصال عناصر می باشد.
- ❖ با بزرگ شدن سیستم ها، افزایش قطعات آنها، طراحی به این صورت، دشوار می شود.
- ❖ یکی از راه کارها برای مقابله با این مشکلات، بهره گیری از زبان های توصیف سخت افزار ( Hardware description language) (HDL) و IC های FPGA (Field programmable gate array) می باشد.

# مفهوم زبان‌های توصیف سخت افزار

- ❖ زبان‌های توصیف سخت افزار، زبان هایی هستند که در آنها قابلیت‌هایی برای مدل‌سازی سخت افزار وجود دارد. به عنوان مثال در زبان Verilog که یکی از این زبان‌ها می‌باشد، دستور and وجود دارد که یک گیت and را پیاده‌سازی می‌کند. یا نوع wire در آن وجود دارد که یک سیم را مدل می‌کند.
- ❖ چند زبان توصیف سخت افزار: Verilog، VHDL، AHDL، AHPL....
- ❖ زبان مورد استفاده در این درس، Verilog می‌باشد. این زبان در حدود سال ۱۹۸۴ میلادی ارائه شد و در چند مرحله در سال های ۱۹۹۵، ۲۰۰۱ و ۲۰۰۵ استانداردسازی شد.

# یک برنامه ساده Verilog

❖ یک برنامه بسیار ساده به Verilog که توضیحات:

❖ کلمه کلیدی module نشان‌دهنده شروع برنامه است. Verilog به هر طرحی که می‌خواهد آن را پیاده کند، یک ماژول می‌گوید.

❖ بعد از کلمه module، یک نام دلخواه برای ماژول، بیان می‌شود. در این مثال، این نام، and\_gate می‌باشد.

❖ سپس، در داخل پرانتز، خروجی‌ها و ورودی‌های مدار، بیان می‌شود. مطابق قرارداد (و نه الزام)، ابتدا خروجی‌ها و بعد ورودی‌ها بیان می‌شوند. به ورودی‌ها و خروجی‌های مدار، اصطلاحاً پورت (Port) می‌گوییم.

❖ در انتهای اکثر دستورات Verilog، سمی کالن، گذاشته می‌شود.

❖ یک گیت and را پیاده می‌کند.

```
module and_gate (c,a,b);  
input a,b;  
output c;  
and a0(c,a,b);  
endmodule
```



## یک برنامه ساده Verilog (ادامه)

❖ یک برنامه بسیار ساده به Verilog که

یک گیت and را پیاده می‌کند.

```
module and_gate (c,a,b);
```

```
input a,b;
```

```
output c;
```

```
and a0(c,a,b);
```

```
endmodule
```

❖ ادامه توضیحات:

❖ در خط دوم و سوم، ورودی یا خروجی بودن Port ها مشخص می‌شود.

❖ خط ۴، دستور اصلی برنامه است. این دستور یک گیت and با ورودی‌های a و b و

خروجی c، ایجاد می‌کند. در اینجا لزوماً باید اول خروجی and و بعد، ورودی‌های and ذکر شوند.

❖ a0 نام گیت and است.

❖ کلمه endmodule (بدون سمی کالن در انتهای آن)، برنامه را پایان می‌بخشد.

# اجرای برنامه Verilog

❖ مشابه زبان‌های نرم‌افزاری، یک برنامه نوشته شده با یک HDL نیز می‌تواند اجرا شود. البته اجرای یک برنامه HDL با اجرای یک برنامه نوشته شده با یک زبان نرم‌افزاری (مانند C++)، متفاوت است. نتیجه اجرای یک برنامه HDL یکی از موارد زیر است.

❖ شبیه سازی (Simulation).

❖ سنتز (Synthesis).

❖ برنامه ریزی (Programming).

## اجرای برنامه Verilog (ادامه)

❖ شبیه‌سازی (Simulation): به معنای ارائه یک سری نمودارهای زمانی (Timing diagram)، از طرح است.

❖ سنتز (Synthesis): ارائه یک شکل (شماتیک) از مدار است.

❖ برنامه‌ریزی (Programming): انتقال طرح به FPGA می‌باشد. FPGAها، Icهایی هستند که شامل تعداد زیادی عنصر منطقی (گیت، عناصر حافظه، mux...) است که اتصالات بین آنها قابل تغییر و برنامه‌ریزی است. برنامه‌های Verilog می‌توانند به فرمت قابل فهم برای FPGA تبدیل شود و به FPGA منتقل شوند. در نتیجه فرد با نوشتن یک نرم افزار یک سخت افزار، تحویل می‌گیرد.

❖ ابزارهای شبیه سازی و سنتز، بیشتر برای تشخیص و رفع اشکالات برنامه کاربرد دارند.

## اجرای برنامه Verilog (ادامه)

---

❖ نرم افزارهایی وجود دارند که یک، دو و یا هر سه کار را انجام می دهند. مثلا Modelsim، قادر به شبیه سازی برنامه های نوشته شده با Verilog یا VHDL است. یا نرم افزار ISE می تواند هر سه کار را انجام می دهد.

# شبیه‌سازی با Modelsim

- ❖ برای انجام شبیه‌سازی با نرم افزارهای شبیه‌ساز مانند Modelsim می‌بایست مقدماً قسمتی که اصطلاحاً Testbench نامیده می‌شود، به برنامه اضافه گردد.
- ❖ در Testbench برای ورودی‌ها در لحظات مشخص، مقادیری در نظر گرفته می‌شود. شبیه‌ساز با توجه به مقادیر داده شده، خروجی را مشخص می‌کند.
- ❖ با ملاحظه مقادیر خروجی مشخص شده توسط شبیه‌ساز، می‌توان خطاهای احتمالی در برنامه را تشخیص داد و برای رفع آنها، اقدام کرد.
- ❖ در Verilog، Testbench تحت قالب یک ماژول مستقل، پیاده‌سازی می‌شود.

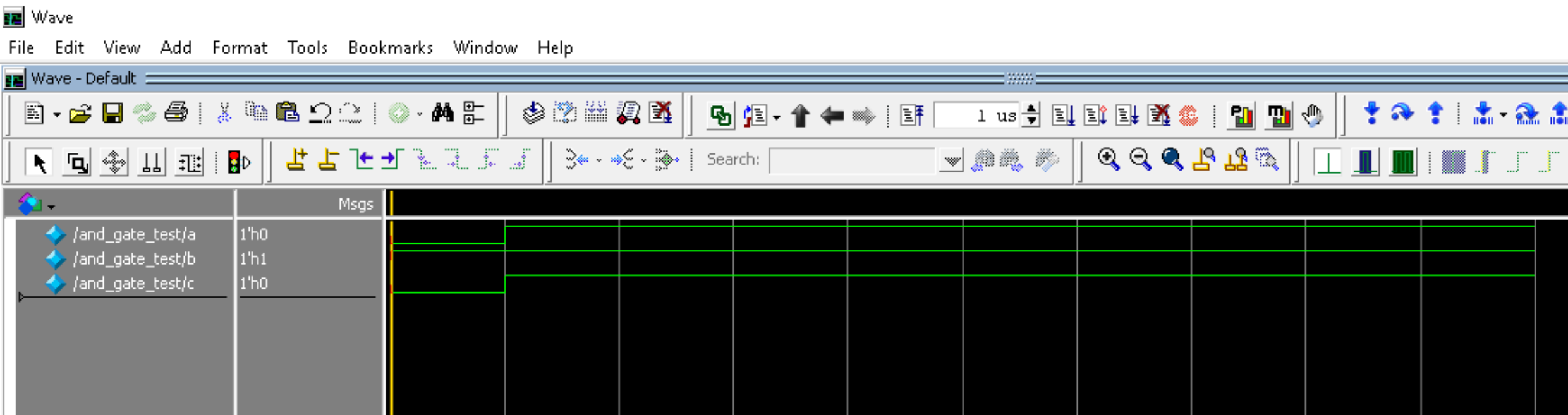
## شبیه سازی با Modelsim (ادامه)

```
module and_gate (c,a,b);  
input a,b;  
output c;  
and a0(c,a,b);  
endmodule  
module and_gate_test;  
reg a,b;  
wire c;  
and_gate ag0(c,a,b);  
initial  
begin  
a=0;
```

```
b=1;  
#100;  
a=1;  
end  
endmodule
```

توضیح: در ماژول تست گفته شده است که شبیه ساز، مقادیر ورودی‌ها را در لحظه صفر برابر  $a=0$  و  $b=1$  در نظر بگیرد. در لحظه  $t=100$  (نانو ثانیه ns)،  $a=1$  بشود.

# شبیه سازی با Modelsim (ادامه)



## سنتز (Synthesis)

---

❖ عمل سنتز را با استفاده از نرم افزار ISE انجام می دهیم.

❖ عمل سنتز توسط این نرم افزار (و سایر نرم افزارهای سنتزگر مانند Leonardo)، به دو شیوه زیر انجام می گیرد.

❖ شماتیک RTL (RTL schematic): یک شکل کلی بر مبنای کتابخانه های موجود در نرم افزار، ارائه می گردد.

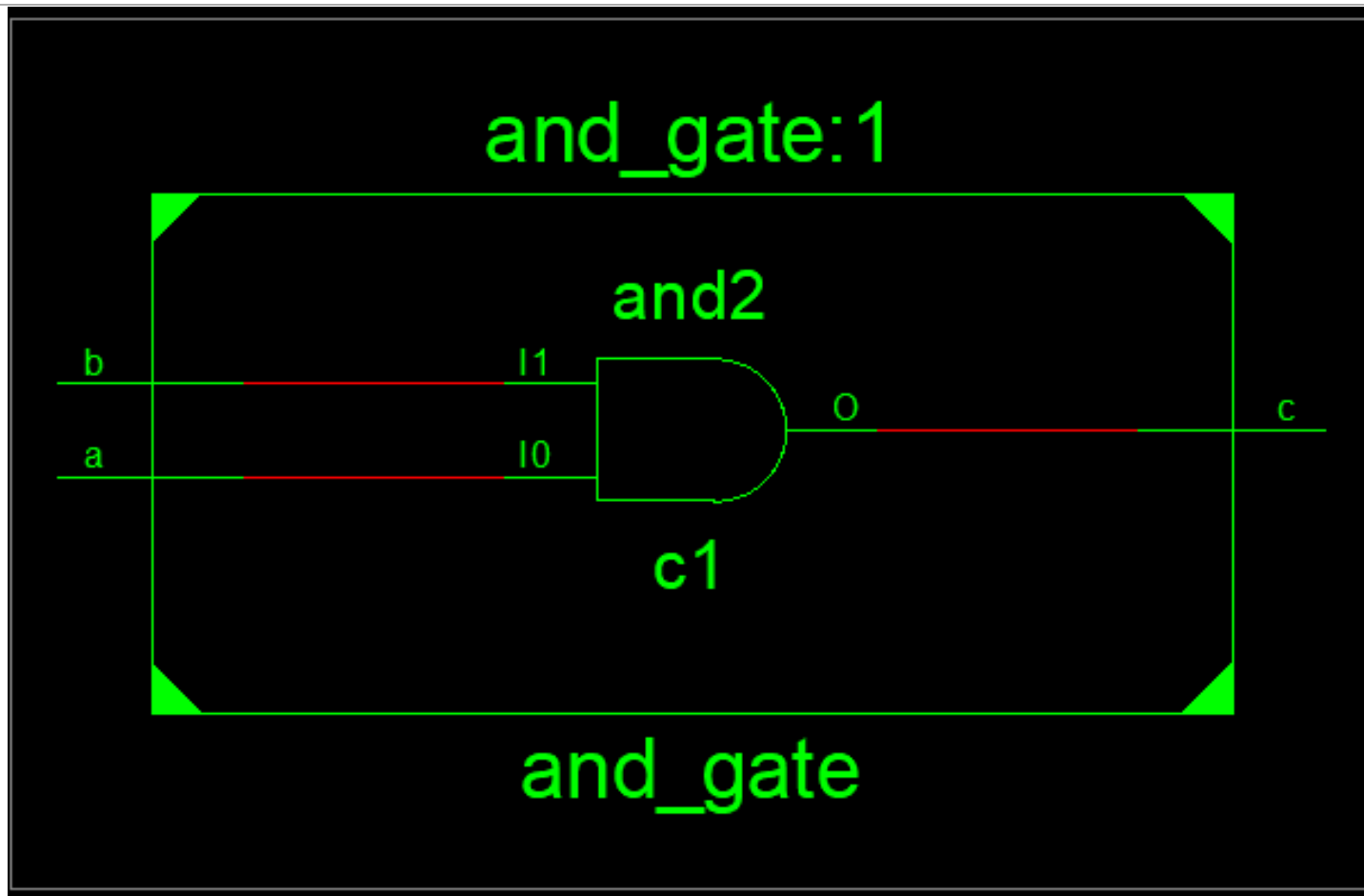
❖ شماتیک مبتنی بر تکنولوژی (Technology schematic): با استفاده از قطعات FPGA ای که ما برای نرم افزار مشخص کرد ایم، شکلی از طرح را نمایش می دهد.



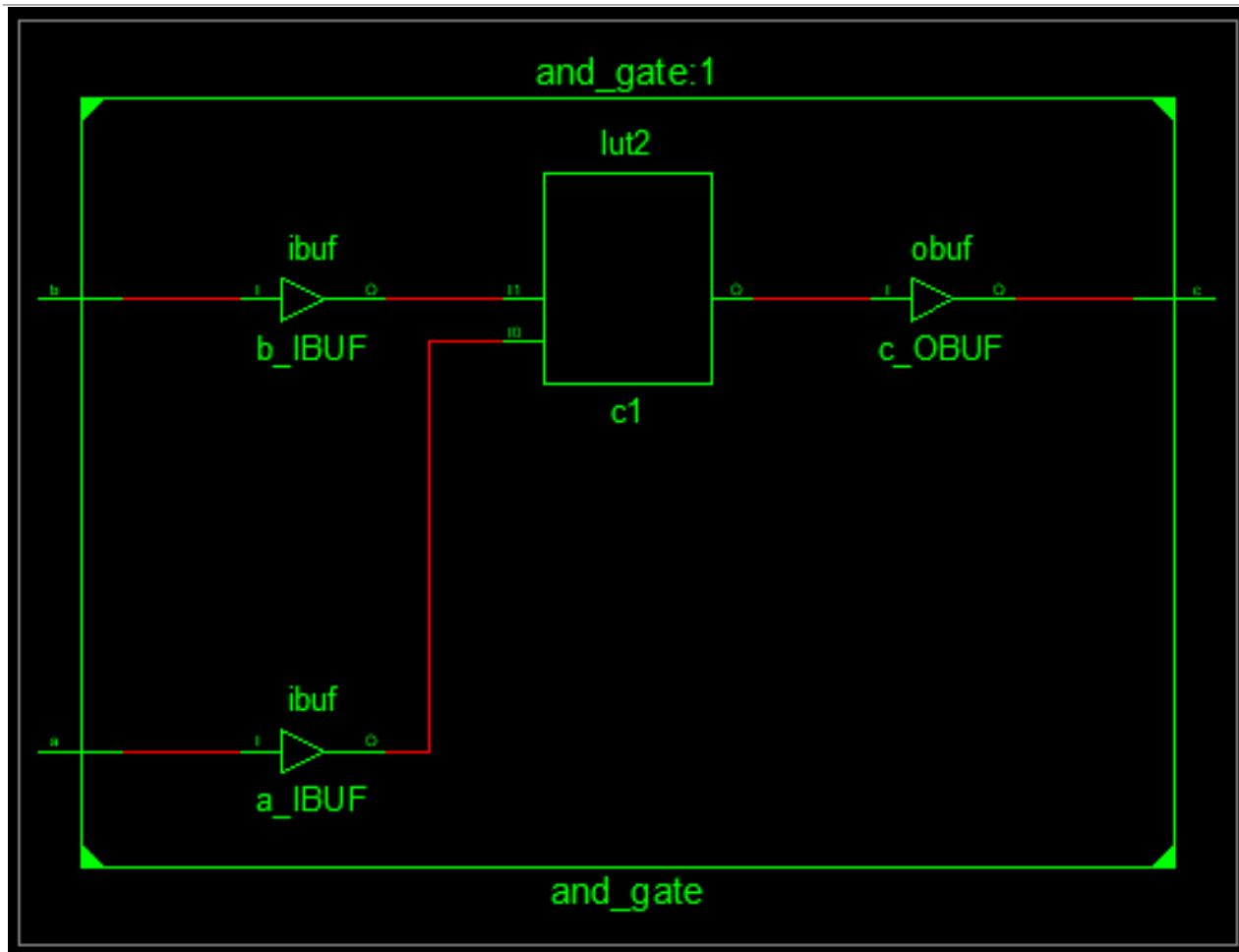
## سنتز (Synthesis) (ادامه)

❖ شماتیک RTL مربوط به

مثال گیت and



## سنتز (Synthesis) (ادامه)



❖ شماتیک تکنولوژی مربوط به مثال  
گیت and

## برنامه ریزی (انتقال طرح به FPGA)

❖ تراشه FPGA به همراه لوازم سخت‌افزاری مربوطه، در یک برد به نام برد FPGA (FPGA board) قرار دارد.

❖ بعضی از این لوازم سخت‌افزاری عبارتند از:

❖ **واحد تأمین کلاک:** وظیفه تأمین سیگنال کلاک FPGA را بر عهده دارد.

❖ **حافظه ROM:** اتصالات درون FPGA با قطع جریان برق، ریست می‌شوند. لذا طرح مربوطه در یک حافظه

ROM قرار می‌گیرد. هنگام روشن شدن برد، طرح به طور خودکار از ROM به FPGA منتقل می‌گردد.

❖ **پورت‌های مربوط به دسترسی به پین‌های FPGA:** پایه‌های FPGA بسیار ریز هستند و کار به طور مستقیم با

آنها مشکل است. لذا پورت‌های متصل به FPGA با پایه‌های بزرگتر وجود دارد که به FPGA وصل هستند و ما

می‌توانیم با آنها کار کنیم.

## برنامه ریزی (انتقال طرح به FPGA) (ادامه)

❖ دیود نوری (LED) (Light emitting diode) و نمایشگر ۷ قسمتی (SSD) (Seven segment display):

این قطعات به بعضی پایه‌های FPGA متصل می‌باشند. در صورتی که در طرح خود این پایه‌ها را به عنوان خروجی انتخاب کنیم، با خاموش/روشن شدن LED و یا بخش مربوطه در SSD می‌توانیم متوجه یک یا صفر بودن خروجی بشویم.

❖ پورت‌های مربوط به انتقال داده به FPGA: مانند پورت USB، موازی، JTAG...

❖ سایر قطعات: مانند کلیدهای خاموش/روشن کردن برد، خازن‌ها، Jumperها...

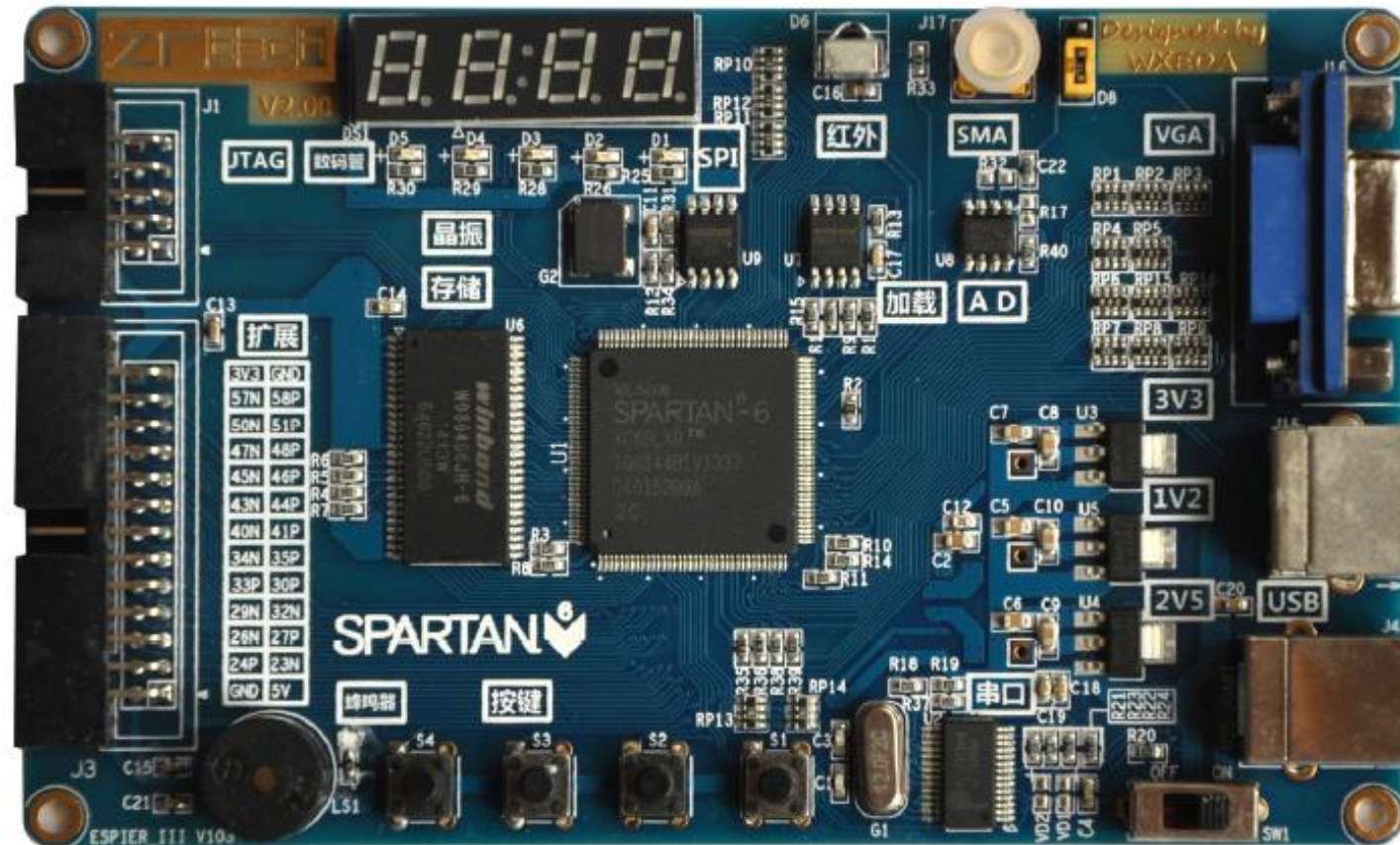
## برنامه ریزی (انتقال طرح به FPGA) (ادامه)

---

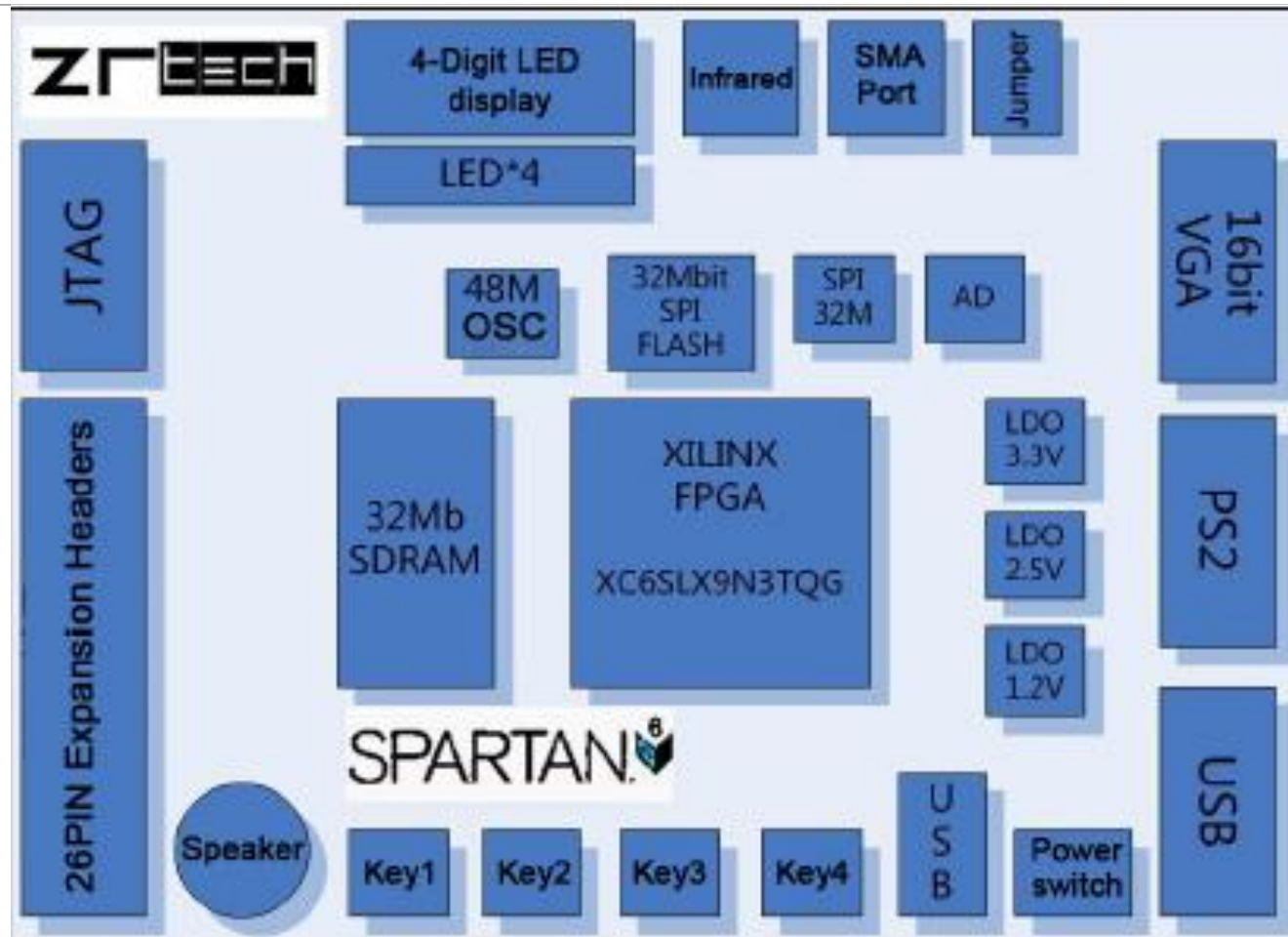
❖ برد FPGA مورد استفاده در این درس، easyFPGA-Spartan-6 نام دارد که ساخت شرکت LogiFind است.

❖ ابتدا توضیح مختصری در خصوص اجزای موجود در برد خواهیم داشت و سپس، نحوه انتقال برنامه Verilog را به این برد را توضیح خواهیم داد.

# برنامه ریزی (انتقال طرح به FPGA) (ادامه)



## برنامه ریزی (انتقال طرح به FPGA) (ادامه)

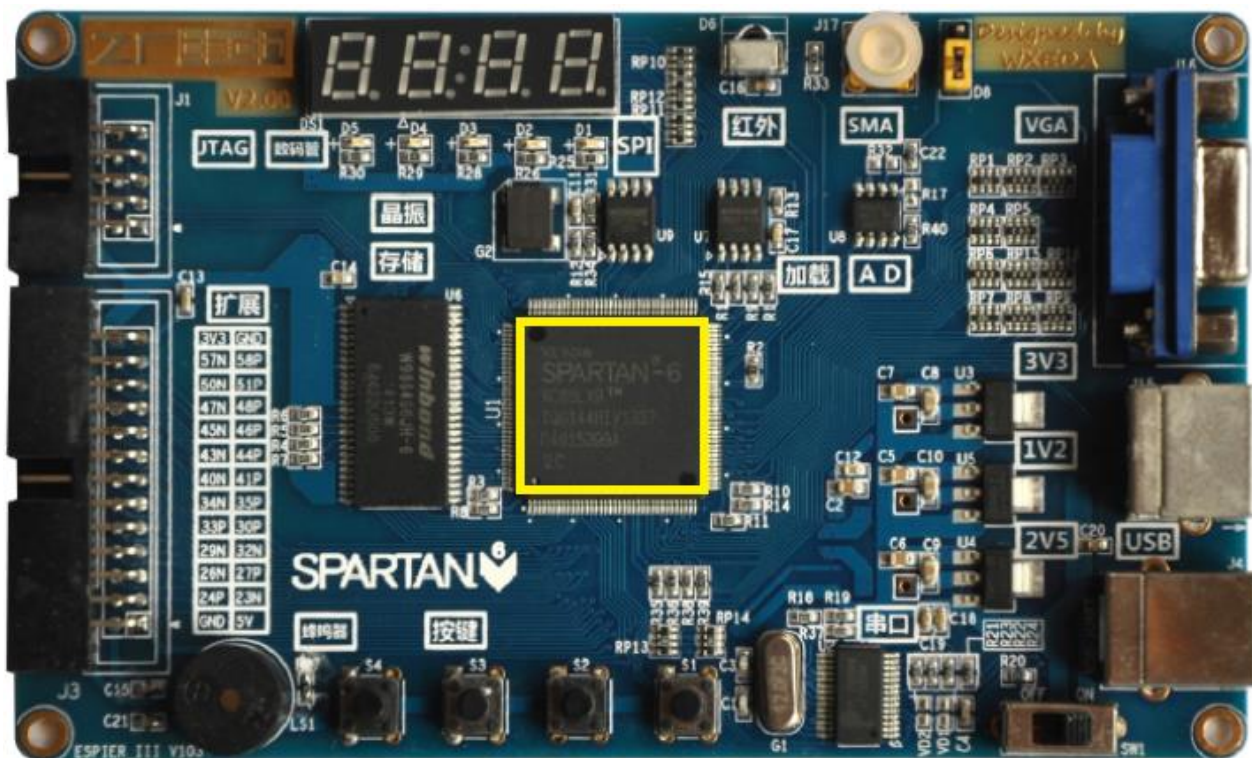




## برنامه ریزی (انتقال طرح به FPGA) (ادامه)

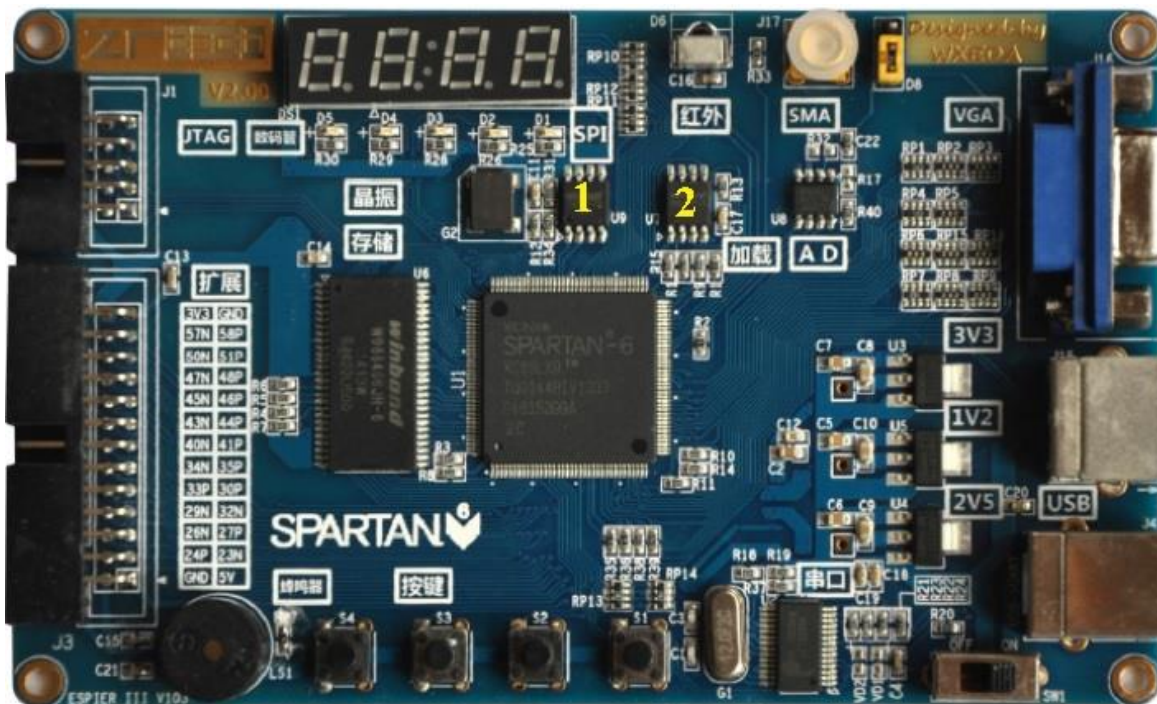
❖ قطعه شماره ۱، FPGA مورد استفاده در برد

است که مهم‌ترین قطعه برد می‌باشد. FPGA مورد استفاده xc6slx9-tqg144 مدل Spartan-6 ساخت شرکت xilinx می‌باشد. عدد ۱۴۴، نشان‌دهنده تعداد پایه‌ها (pins) تراشه می‌باشد (تراشه xc6slx9 به سه صورت ۱۴۴، ۱۹۶، ۲۲۵، ۲۵۶ و ۳۲۴ پایه‌ای عرضه شده است که FPGA موجود در برد، از نوع ۱۴۴ پایه‌ای می‌باشد.





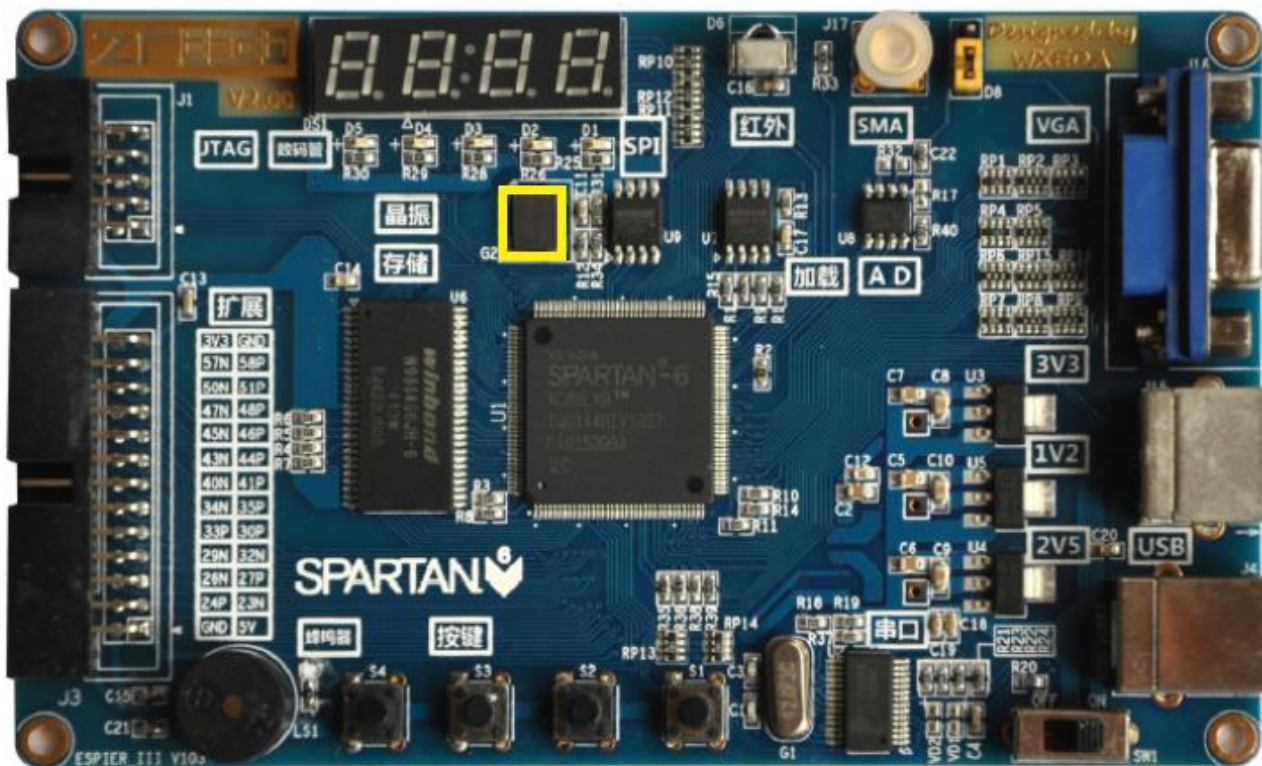
## برنامه ریزی (انتقال طرح به FPGA) (ادامه)



❖ تراشه شماره ۱، یک flash ROM سری، با ظرفیت ۶۴Mbit است. زمانی که برق FPGA قطع می‌شود، محتوی آن پاک می‌شود. لذا، معمولاً طرح را در داخل ROM قرار می‌دهند که با قطع جریان برق، پاک نشود. تنظیمات، طوری انجام می‌شود که هر زمان که مورد روشن شد، طرح به طور خودکار از ROM به FPGA برود.

❖ Flash ROM به صورت سری است و بر مبنای استاندارد SPI (Serial peripheral interface) عمل انتقال را انجام می‌دهد. تراشه شماره ۲ واسط SPI خوانده می‌شود و این انتقال سری را مدیریت می‌کند.

## برنامه ریزی (انتقال طرح به FPGA) (ادامه)



❖ IC نشان داده شده در شکل، یک

اسیلاتور (Oscillator) می باشد.

❖ اسیلاتور، قطعه ای است که سیگنال

کلاک لازم برای عملکرد بورد FPGA را

تأمین می کند.

❖ فرکانس کلاک تولید شده توسط این

قطعه، ۴۸MHz می باشد.

# برنامه ریزی (انتقال طرح به FPGA) (ادامه)

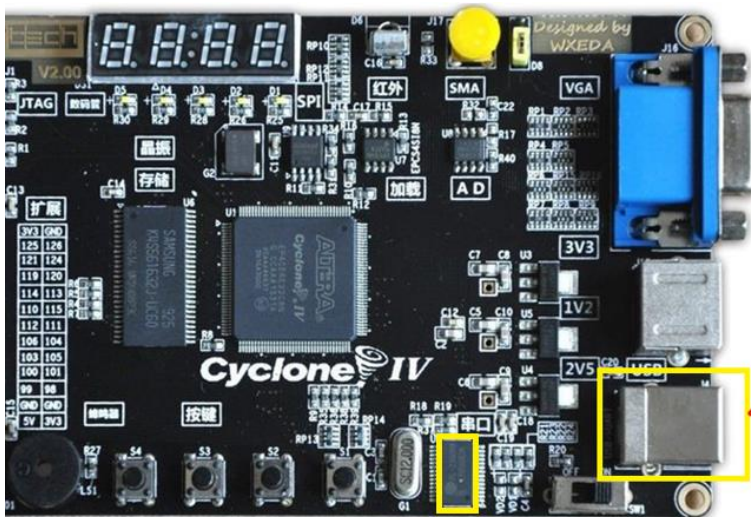
❖ USB → UART

❖ UART (universal asynchronous receiver-transmitter)، یک ابزار سخت‌افزاری برای انتقال سری ناهم‌گام (Asynchronous serial communication) استفاده می‌شود.

❖ USB نیز یک واسط پیشرفته‌تر برای انتقال سری است.

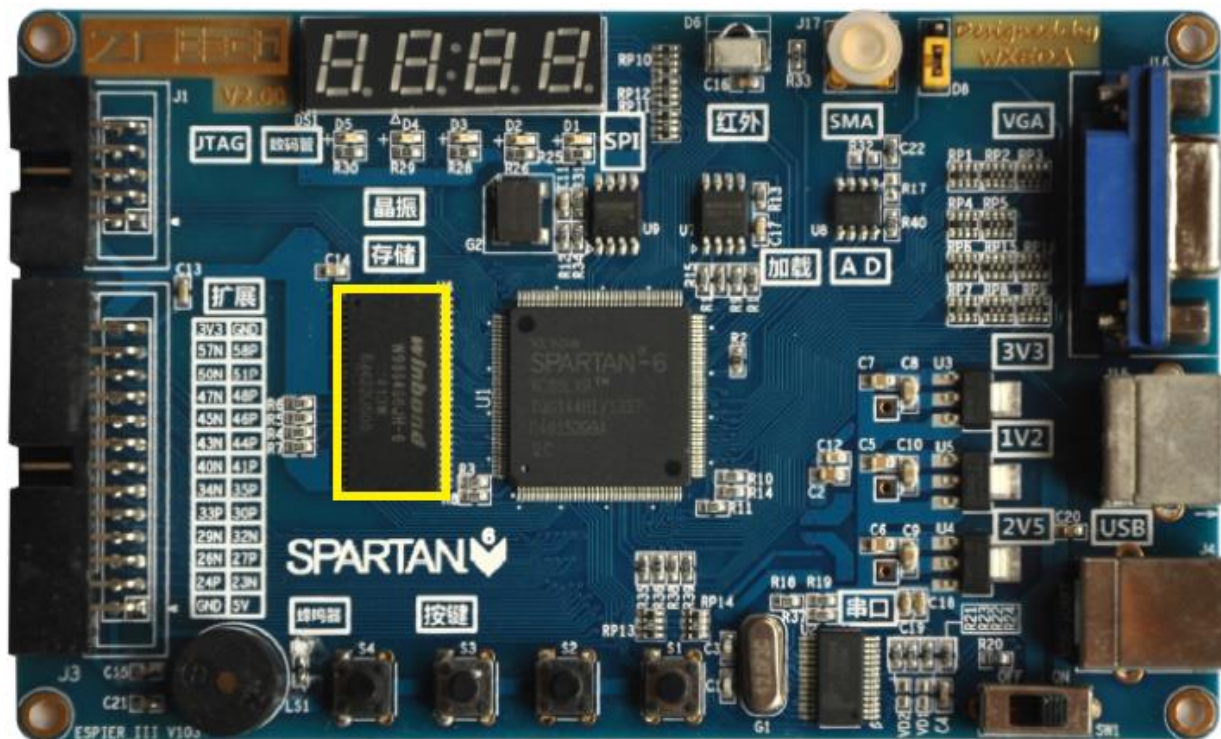
❖ برد مورد استفاده از تکنولوژی UART برای انتقال داده استفاده می‌کند. اما با توجه به اینکه کامپیوترهای امروزی از USB استفاده می‌کنند. لذا وجود مبدل USB->UART ضروری می‌باشد.

❖ مستطیل زرد بزرگ، نشان‌دهنده درگاه ورود داده USB است. مستطیل زرد کوچک، تراشه‌ای است که اصطلاحاً USB->UART bridge خوانده شده و عمل تبدیل را انجام می‌دهد. تراشه مورد استفاده در برد PL2303 نام دارد که از تراشه‌های معروف در این حوزه است.





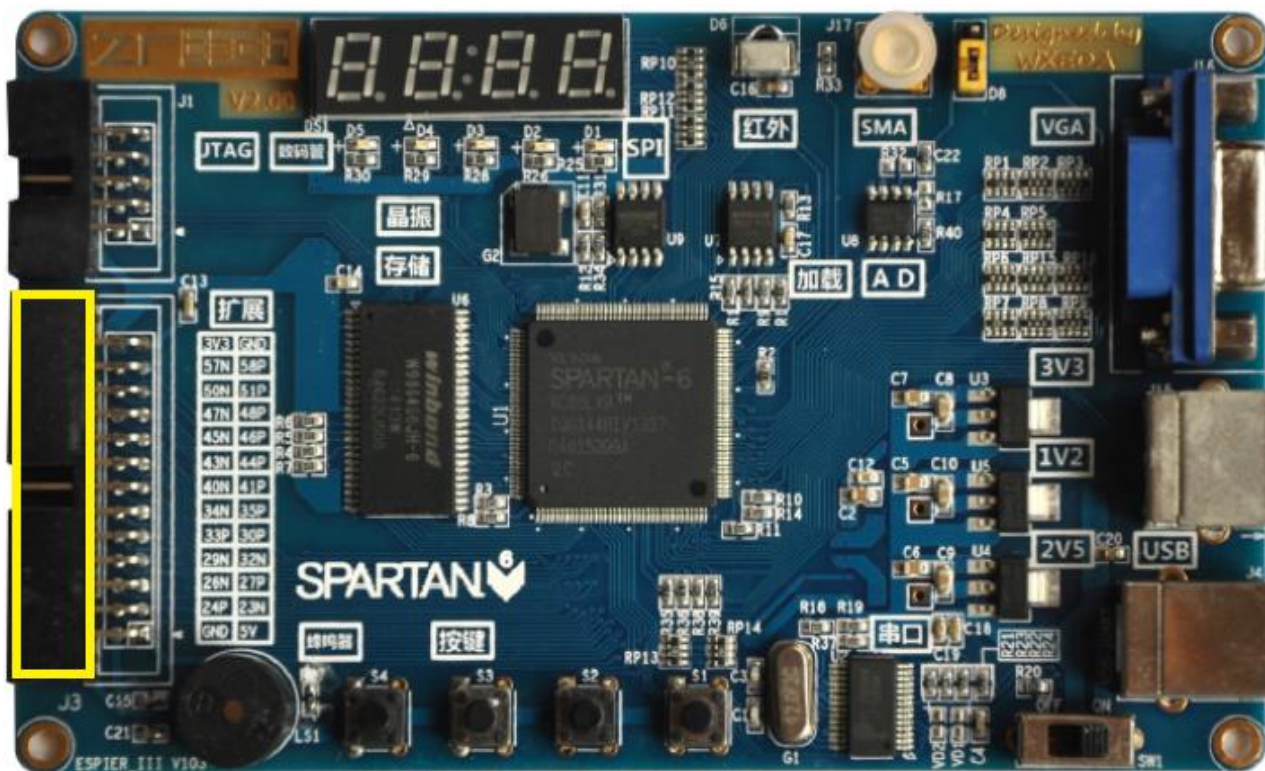
## برنامه ریزی (انتقال طرح به FPGA) (ادامه)



❖ قطعه نشان داده شده در شکل، یک حافظه RAM از نوع SRAM با ظرفیت ۶۴Mb می‌باشد. در انجام عملیات روی برد، این قطعه، نقش یک حافظه موقت (بافر) را بازی می‌کند.

❖ طبیعتاً بر خلاف Flash ROM با قطع جریان برق، محتوی SRAM از بین می‌رود.

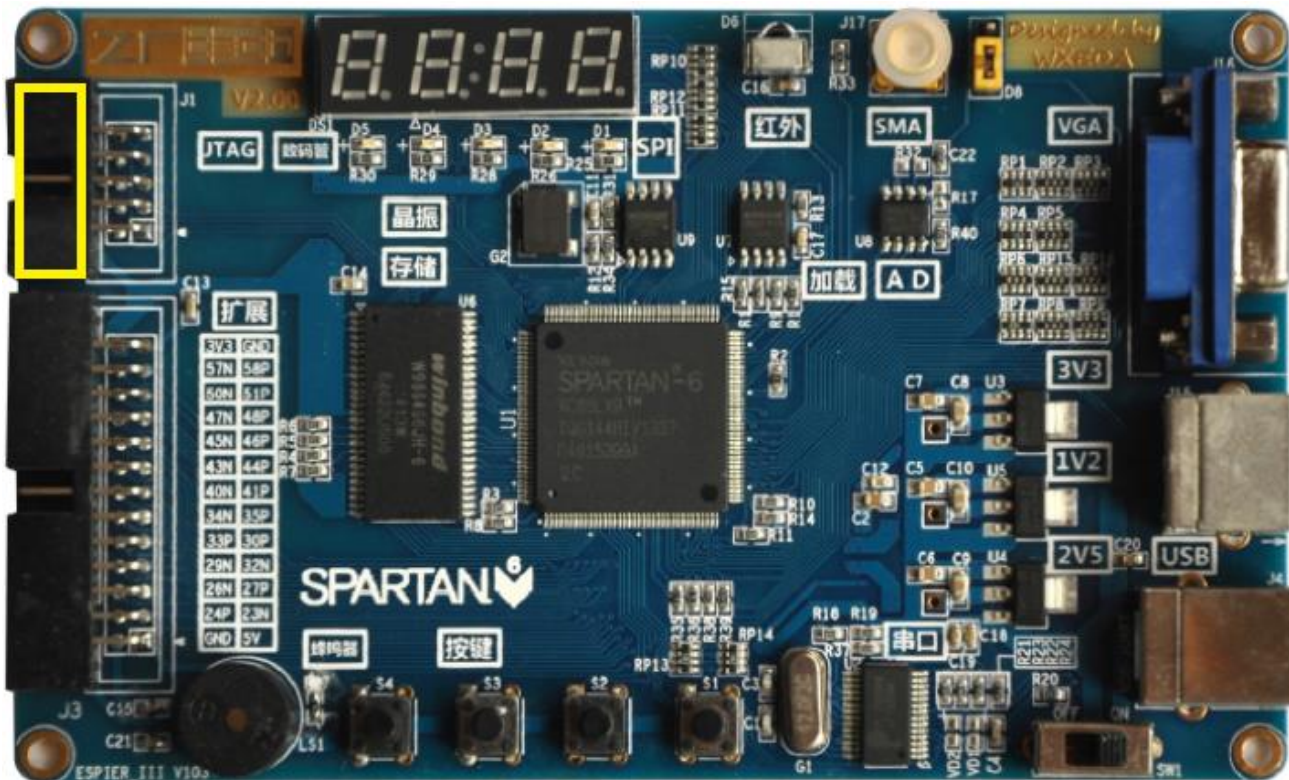
## برنامه ریزی (انتقال طرح به FPGA) (ادامه)



❖ قطعه مشخص شده در روبرو، اصطلاحاً گسترش پایه (Pin expansion)، نامیده می‌شود. پایه‌های FPGA ریز بود و امکان دسترسی به آنها فراهم نیست. برای حل مشکل، تعدادی از پایه‌ها به Pin expansion وصل می‌شوند. کاربر می‌تواند از طریق این قطعه، ورودی مورد نظر خود را به FPGA داده و خروجی خود را نیز از FPGA بگیرد.

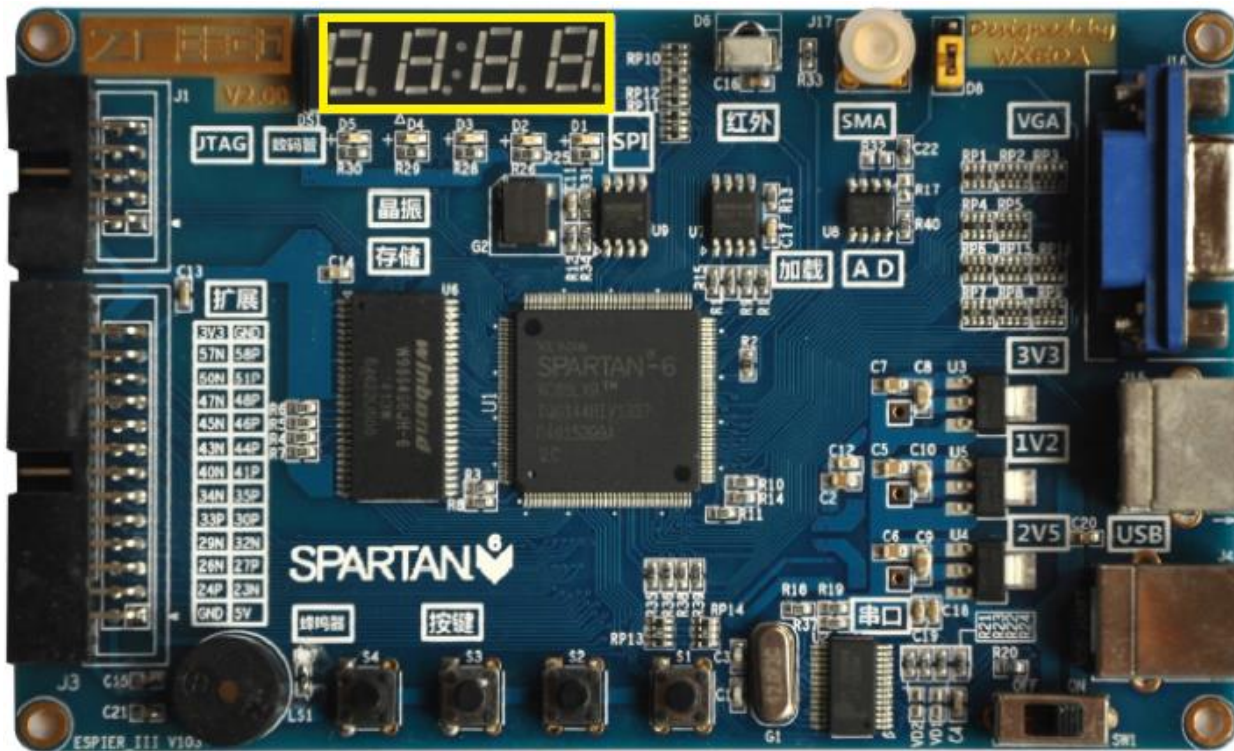


# آشنایی با اجزای برد (ادامه) برنامه ریزی (انتقال طرح به FPGA) (ادامه)



❖ قطعه مشخص شده در روبرو، اتصال JTAG خوانده می شود. یکی از روش های انتقال طرح ها به FPGA استفاده از پورت JTAG می باشد.

## آشنایی با اجزای برد (ادامه)



- ❖ ۴ عدد SSD (Seven segment display) در برد وجود دارند. تعدادی از پایه‌های خروجی FPGA به دیودهای نورانی مربوط به این SSDها وصل شده‌اند.
- ❖ از این SSDها می‌توان برای بررسی عملکرد مدار استفاده کرد. اگر یک خروجی باید یک باشد، دیود وصل شده به آن در SSD بایستی خاموش باشد و در غیر این صورت، دیود باید خاموش باشد.