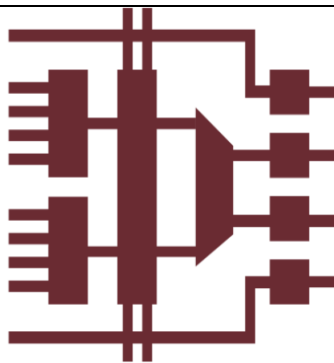




FPGA-SYSTEMS.RU

сообщество FPGA и SoC разработчиков



Китай, GOWIN, АО "Восток", ПЛИС - лучше меньше, да лучше!

Автор: **digitalinvitro**

Рецензент: **KeisN13**

2019 г.



Оглавление

Аннотация	3
Отладочный комплект	3
Тестовый проект.....	6
Заключение	11

Аннотация

То, что Китай научился разрабатывать собственные ПЛИС было мне известно и раньше. Подолгу разглядывая на aliexpress плату разработчика на EG4S20 я взвешивал аргументы в пользу купить и попробовать. Однако, почти сразу, останавливала неизвестность со средой разработки - "а ну как там на чистом китайском иероглифическом языке весь интерфейс будет выполнен?" Благодаря [новости](#), а особенно комментарию от сотрудника [АО "Восток"](#) под ней о том, что "среда вполне себе на дружелюбном английском, да и за лицензией обращаться можно в АО "Восток"" - все сомнения были развеяны и энтузиазм разгорелся с новой силой. Плата разработчика Lichee Tang Nano была куплена по сходной цене, а лицензии на IDE [GOWIN](#) и Synplify практически тут же получены от АО "Восток".

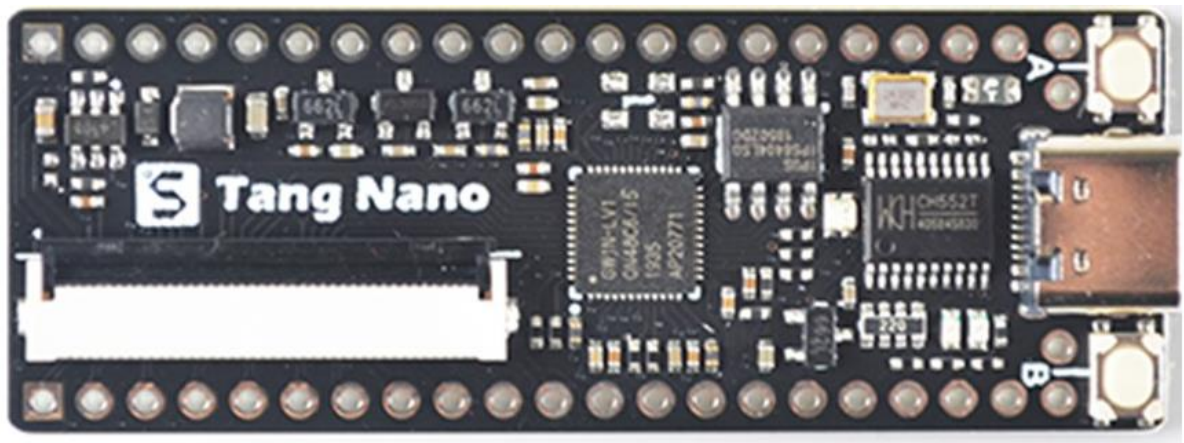
Отладочный комплект

Хотелось бы коснуться самой платы и ПЛИС GW1N-1, на которой она построена. Прежде всего, меня удивила цена отладочной платы при таком размере и содержанием. Тут прекрасно все, как мне кажется, и цена в районе 600-700 рублей за отладочную плату и корпус QFN48 и наличие на борту ПЛИС с 1152 4-х входных LUT, 864 триггера и 9 кбайт(72Кбит) блочного ОЗУ. Именно такого скромного, но, тем не менее, с накристалльным ОЗУ и хотелось видеть. 9К хватит на ПЗУ микрокоманд и загрузочный сегмент ОЗУ. Как говорится "у кого что болит"....

Нечего и говорить о том, что установка среды и лицензий не содержала подводных камней и была довольно дружелюбной. По "приезду" платы на ней тут же был отыгран тест всех времен и народов, моргание светодиодом, которых на плате для нужд пользователя ровно один, но... зато RGB! В целом плата показалась мне сбалансированной по предоставляемым разработчику возможностям, если кратко:

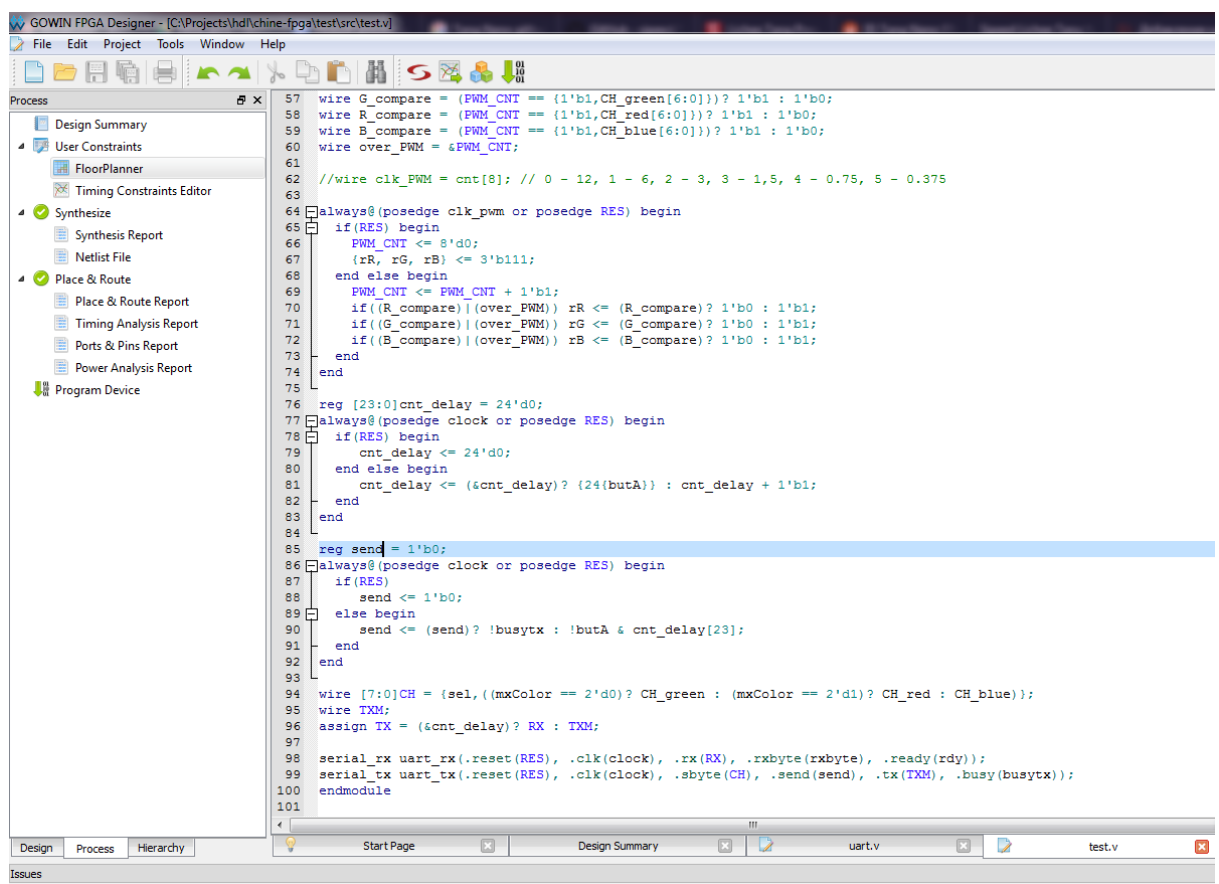
- 2 кнопки, замыкаемые на GND и подтянутые резисторами на питание;
- 3-х цветный RGB светодиод;
- генератор 24МГц;
- мост USB-UART;
- IO, выведенные на разъем для ЖК и параллельно на краевые линейки пинов;
- 64 мбит QSPI-PSRAM

Благодаря малым размерам ПЛИС и комплектующих, разработчику удалось уместить плату в формат DIP-40, что действительно не может не радовать. С подключением платы по USB Type-C можно бы и поспорить, но зачислить это в ранг недостатков рука не подымается. На плату имеется [схема](#) и ресурсы в интернете [с примерами проектов](#) - мигание светодиодами и вывод на LCD, который, кстати, можно было заказать вместе с платой.

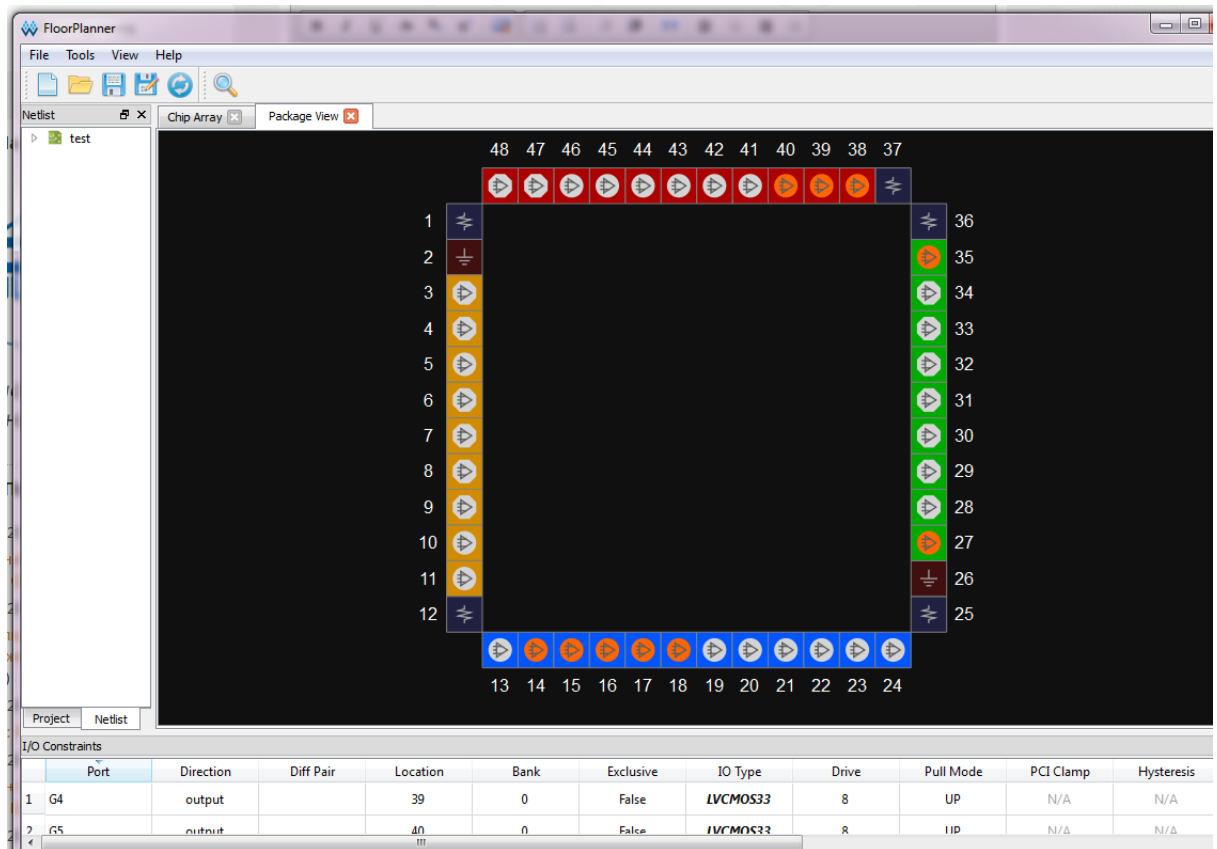


Так выглядит Tang Nano

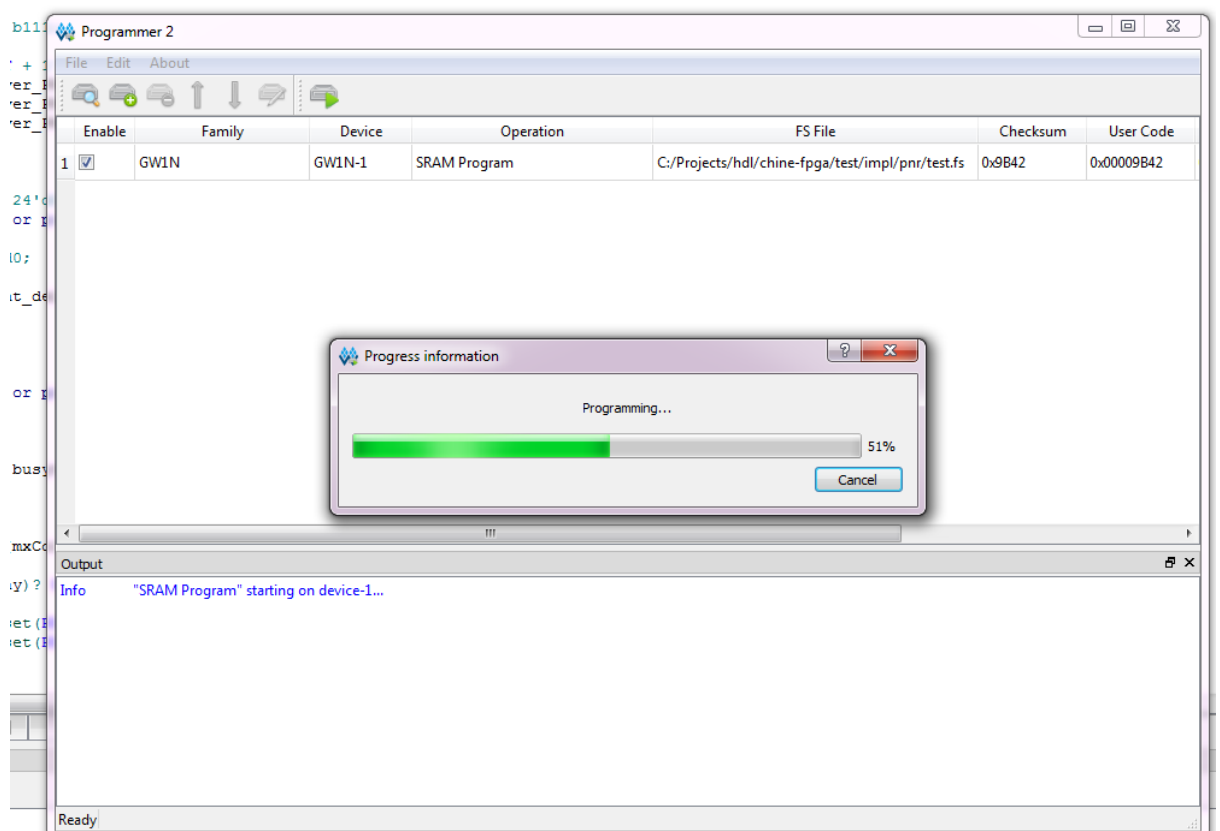
Осилив тест с морганием и увидев, что среда не менее дружелюбна чем ISE или Quartus, можно было попробовать и что-то более сложное.



Внешний вид среды разработки GOWIN



Назначение IO в FloorPlanner



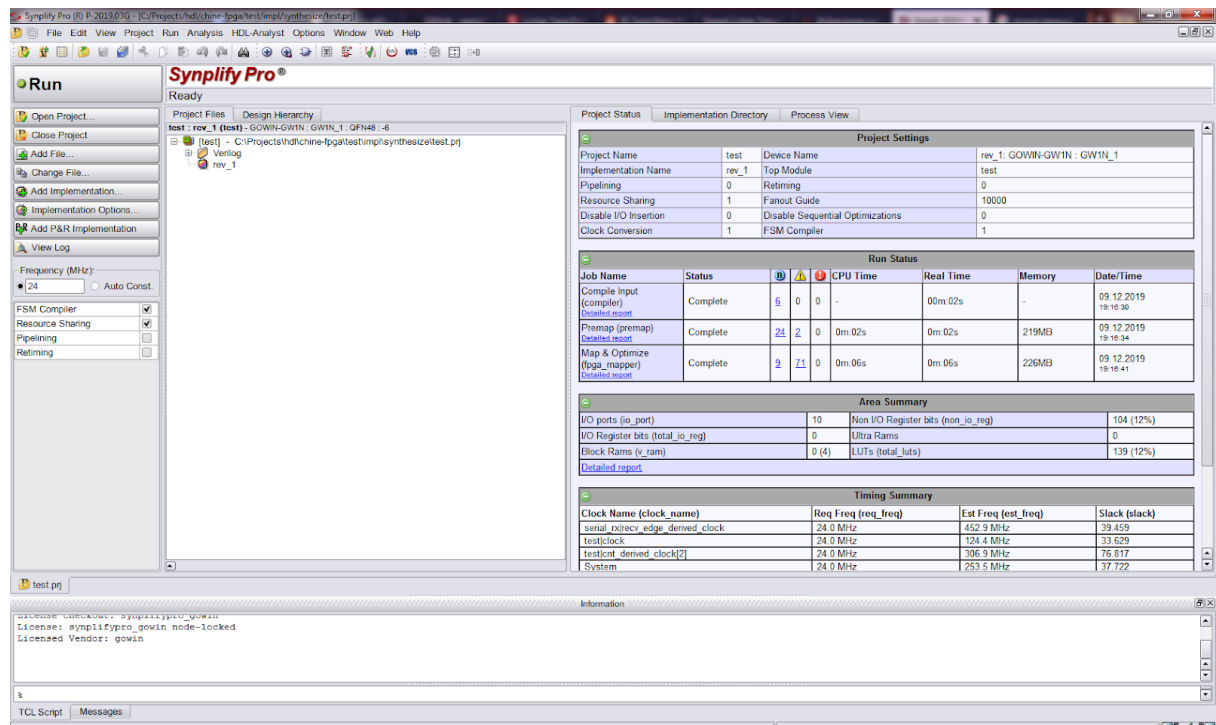
Программатор

Тестовый проект

Дабы далеко не отходить от RGB-светодиода решено было сделать управляемый по UART трехканальный ШИМ с таким расчетом, чтобы и удобно и при этом хватало возможностей терминала. Предположим, что скважность канала цвета можно менять значением принятого в терминал символа. Предположим, также, что есть символ, который является “табуированным” (=) и приводит к переходу из **режима изменения скважности** канала в **режим выбора** канала цвета. Выбор канала осуществляется вводом в терминал одного из символов r, g, b. Получается простенький автомат из двух состояний плюс приемник и передатчик UART. Учитывая входную частоту 24МГц, чтобы не мучиться с большими цепями деления, установим скорость передачи/приема по UART в 115200 бод. По нажатию на кнопку “А” в UART выдается значение текущего канала. На дополнительные пины выдается сигнал текущего канала и состояние триггера выбранного режима. Трехканальный ШИМ, опять же, чтобы не мучиться, со сменой скважности в любой момент счета. Модуль UART был разработан ранее специально для очень маленьких ПЛИС (конкретно для Xilinx xc9572), в которые кроме UART нужно впихивать и многое другое.

И вот тут то... сразу не задалось, первая же попытка “пощупать” за UART, организовав простейший байпас с RX на TX, провалилась, хотя казалось чего бы уж проще? Поплутав некоторое время в назначении системных ног ПЛИС (RECONFIG_N и DONE) для каналов UART и поняв с помощью подсказок от АО “Восток”, что искать надо не там где светло, а там где темно и лениво, вооружился осциллографом и выяснил, что часть микросхемы CH552T, ответственная за UART, на поверку “сопела в тряпочку” подняв ноги RX/TX к питанию. Вообще данная микросхема вела себя крайне подозрительно! Выяснилось, что детектировать ПЛИС по JTAG, находящийся в этом же мосте, она желает не на всех скоростях (только 2МГц) и ни при каждом запросе из программатора. При этом если воспользоваться настройками среды для прошивки, попадающими в программатор автоматически при его вызове, ПЛИС прошивалась всегда и на любой из имеющихся скоростей. Отнеся эти особенности к национальным, я решил больше уже не мучить CH552T, который, кстати сказать, в системе определился без драйверов в виду его родства толи с CH340G толи с FTDI, чьи драйвера в системе уже имелись. Как бы не было горько, от осознания несамодостаточности Tang Nano, пришлось прицепить к плате внешний USB-UART и продолжить эксперимент.

Начиная с этого момента “вечер перестал быть томным”, китаец начал показывать свой норов, а синтезатор вывалил кучу предупреждений, часть из которых, надо отдать должное, была исключительно по существу вопроса. Как уже, наверное, стало понятно китаец для синтеза и размещения использует Synplify.



Внешний вид среды разработки в Synplify Pro

Оказалось, что IDE GOWIN констрейнты user clock по каким-то своим причинам игнорирует и Synplify честно сообщает об этом. Избежать этого можно перейдя в Synplify и подключив констрейнты в его формате (tcl). Проект можно собрать и оттуда, но кроме клона было много чего еще. Например, синтезатор не хотел мириться с тем, что где то в проекте инициализация триггера производится одним значением, а асинхронный сброс устанавливает противоположное значение триггера.

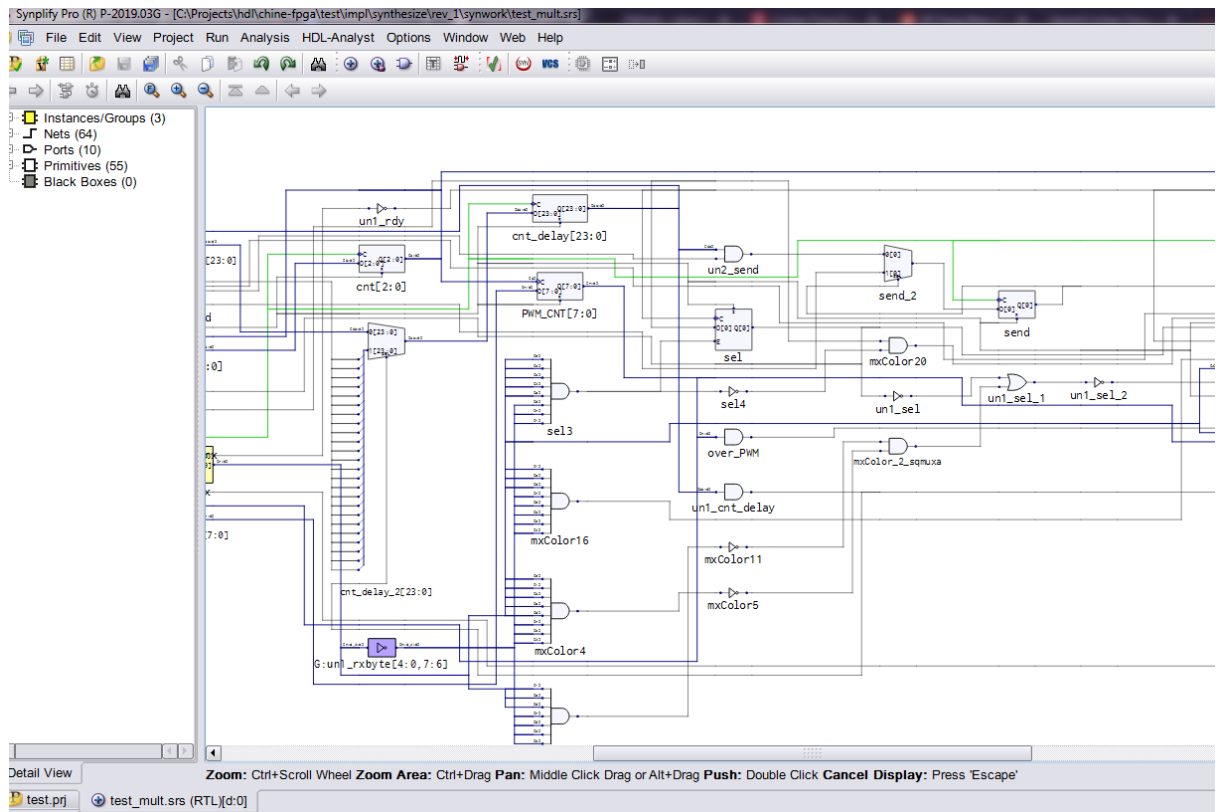
```

reg send = 1'b0;
always@(posedge clock or posedge RES) begin
    if(RES)
        send <= 1'b1;
    else begin
        send <= (send)? !busytx : !butA & cnt_delay[23];
    end
end
  
```

@W: BN680 : "c:\projects\hdl\chine-fpga\test\src\test.v":86:0:86:5|Found conflicting initial value on the register send. Possible simulation mismatch. Specified technology does not support initial value of 0 on register with asynchronous set. To resolve, update the RTL with correct initial value.

Согласен и не буду спорить - это логичное требование, но иногда практикую инициализацию триггера одним значением, чтобы потом сбросом посмотреть изменения. Всегда воспринимал инициализацию триггеров, как этап конфигурирования ПЛИС "битстримом" из

конфигурационного флеша. GOWIN делает это путем асинхронного сброса триггеров там, где он прописан.



На схему тоже можно посмотреть



И даже при сформированном файле *.rosp увидеть внутренности



Навигация в ChipArray

Худо-бедно проект заработал, но не стабильно. Очень долго я не мог понять, зная, что на Xilinx и Altera все работало, что же происходит?! Не обращал внимание на то, что синтезатор подсказывает мне в чем дело, так как смирился с тем фактом, что IDE не видит констрейнта на клок, а значит, думал я, не понимает, что остальные клоки в модулях - производные, вот и жалуется на них! Таким образом, я прохлопал сообщение, что использую "загейтованный" клок в ресивере. Сделал необходимое в таких случаях - ушло замечание при синтезе и появилась стабильность приема по UART.

```
Mapping to part: gw1n1qfn48-6
Cell usage:
ALU          48 uses
DFFC         62 uses
DFFCE        8 uses
DFFP         9 uses
DFFPE        25 uses
GSR          1 use
INV          3 uses
MUX2_LUT5    7 uses
LUT2         22 uses
LUT3         53 uses
LUT4         64 uses

I/O ports: 10
I/O primitives: 10
IBUF         4 uses
OBUF         6 uses

I/O Register bits: 0
Register bits not including I/Os: 104 of 864 (12%)
Total load per clock:
  test|clock: 55
  test|cnt_derived_clock[2]: 14

@S |Mapping Summary:
Total LUTs: 139 (12%)

Mapper successful!

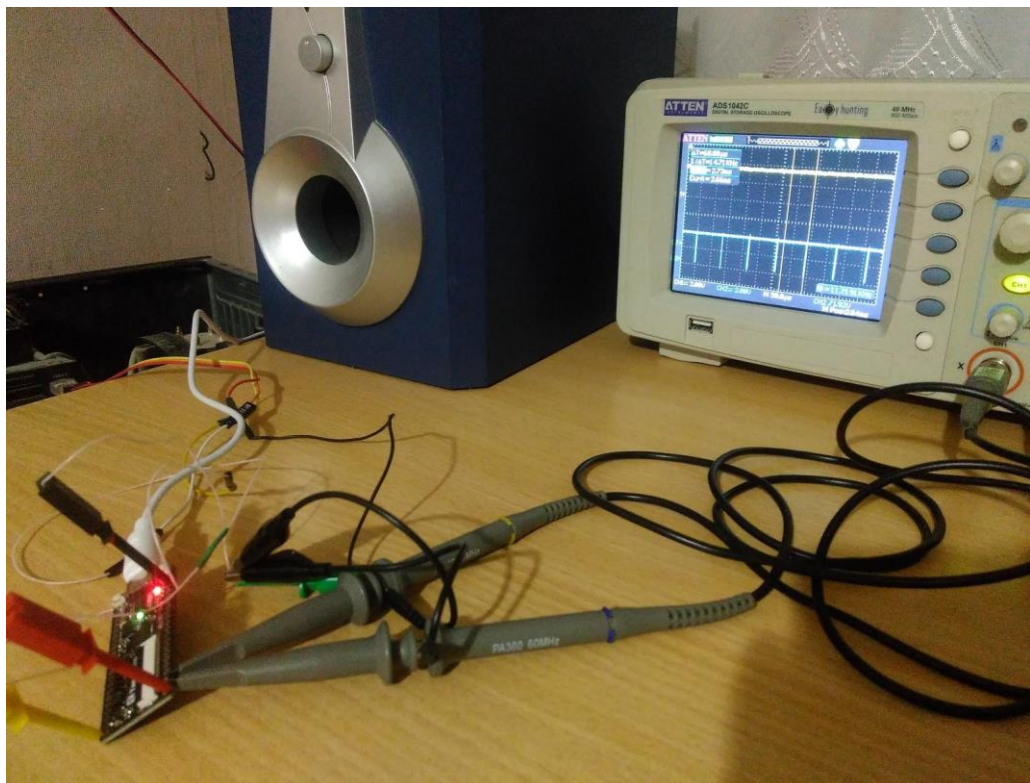
At Mapper Exit (Real Time elapsed 0h:00m:06s; CPU Time elapsed 0h:00m:05s; Memory used c

Process took 0h:00m:06s realtime, 0h:00m:06s cputime
# Mon Dec 9 19:16:41 2019

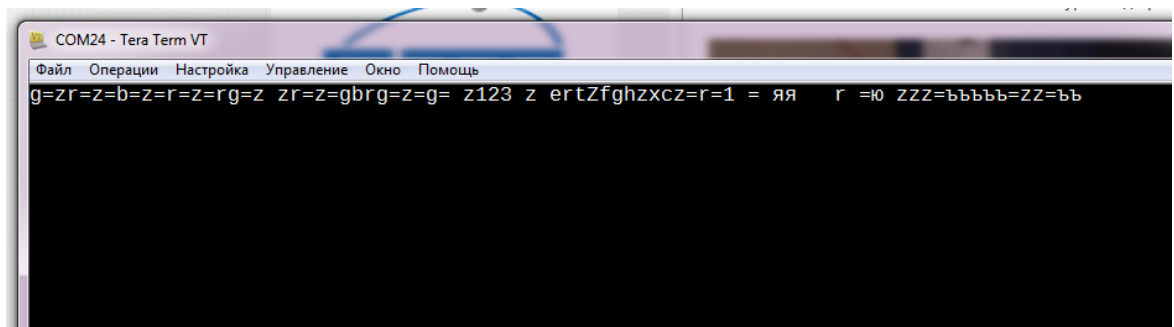
#####]

Design Summary  x  uart.v  x  test.v  x
```

Ресурсы под проект



Проект работает, все довольны, все свободны :)



Настройка каналов в терминале

Заключение

Резюмируя, хотелось бы сказать, у китайцев получилось и получилось весьма недурно! Конечно, сравнивать ISE и Quartus (Xilinx/Altera) с IDE GOWIN будет некорректно учитывая, что последний прибегает к помощи третьих лиц (Synplify Pro © Synopsys), но можно вполне корректно сравнивать стоимость решений и богатство имеющихся при такой стоимости аппаратных возможностей.

P.S. Выражаю благодарность работникам [АО “Восток”](#) за доставленное удовольствие от работы с GOWIN и бесценную консультационную помощь.

Исходные файлы проекта <https://github.com/digitalinvitro/Tang-Nano-test>

Понравилась статья? Не забудьте поддержать автора

