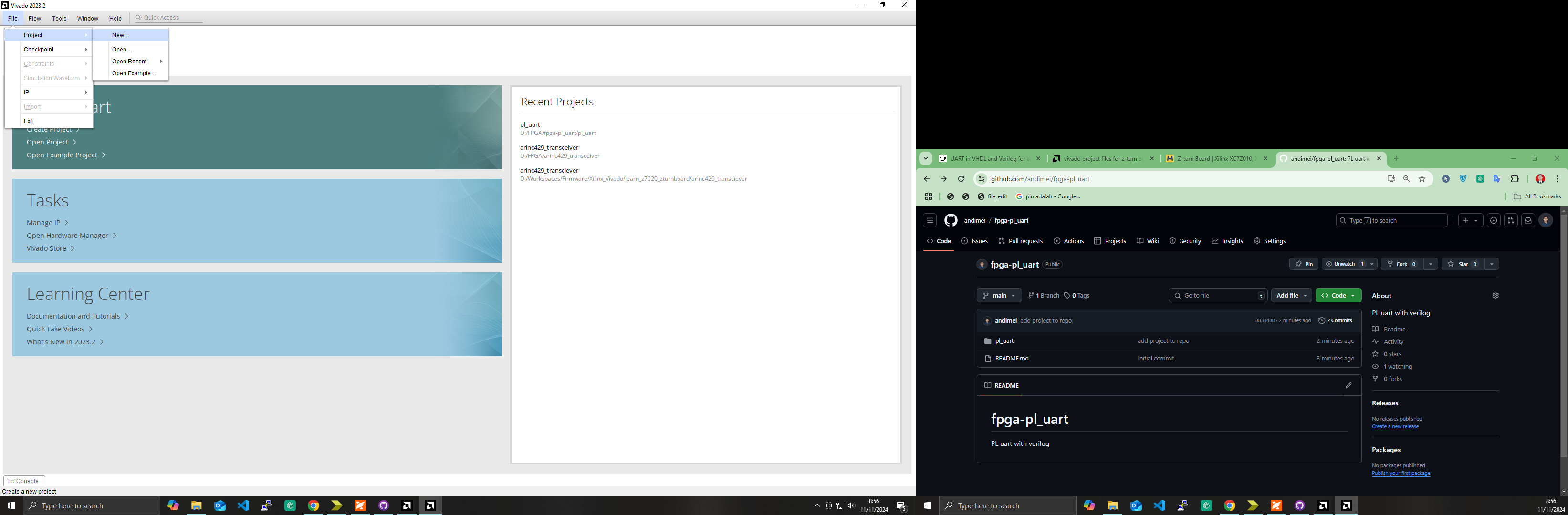
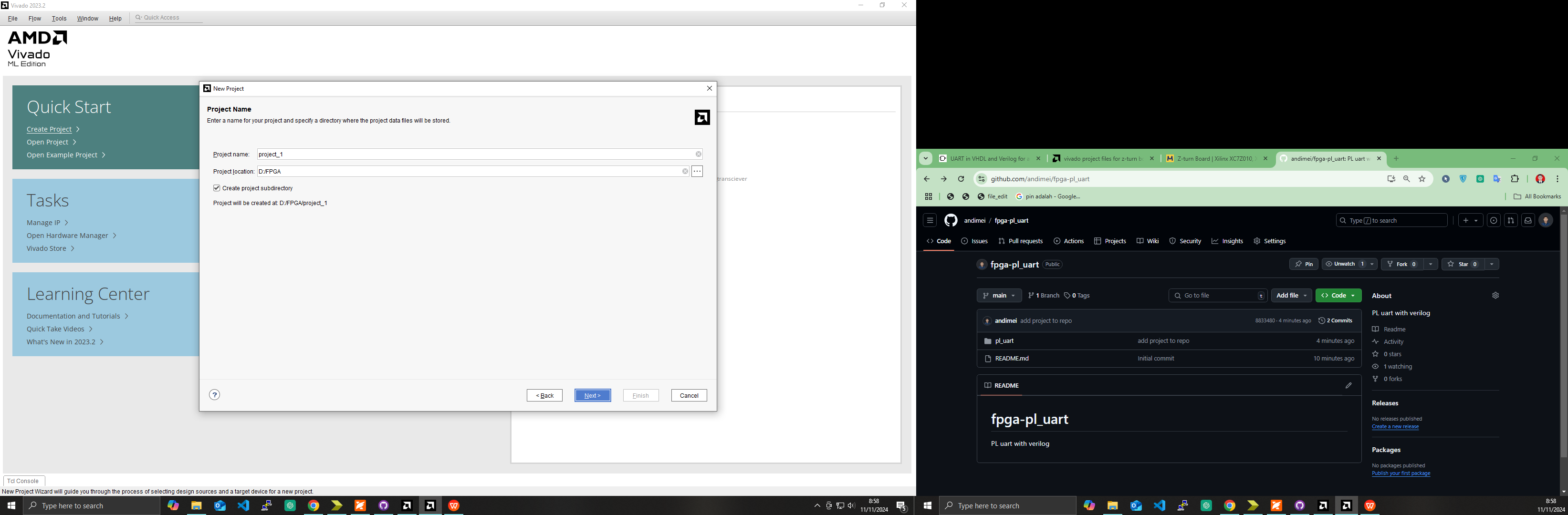
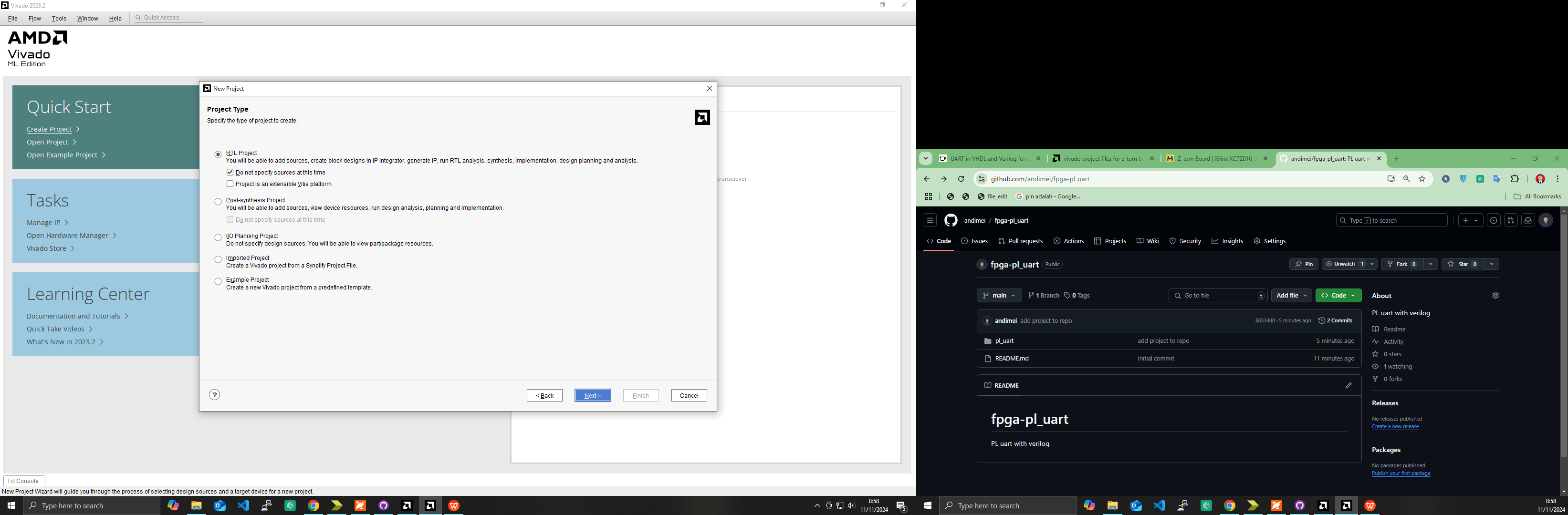
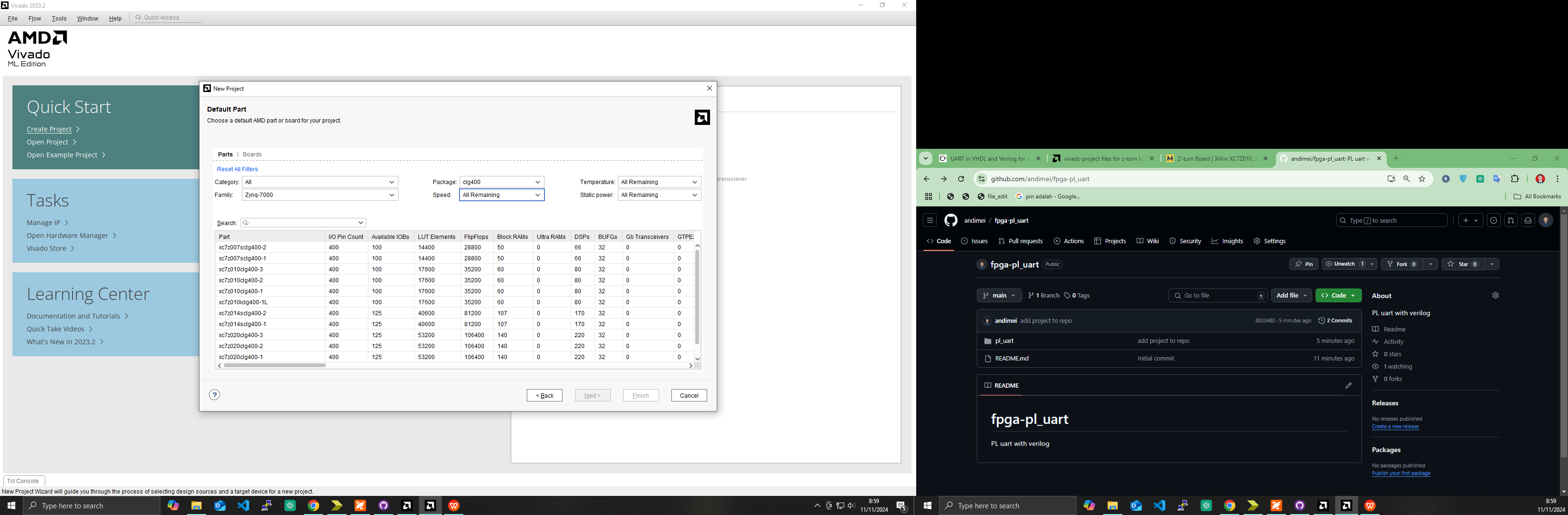
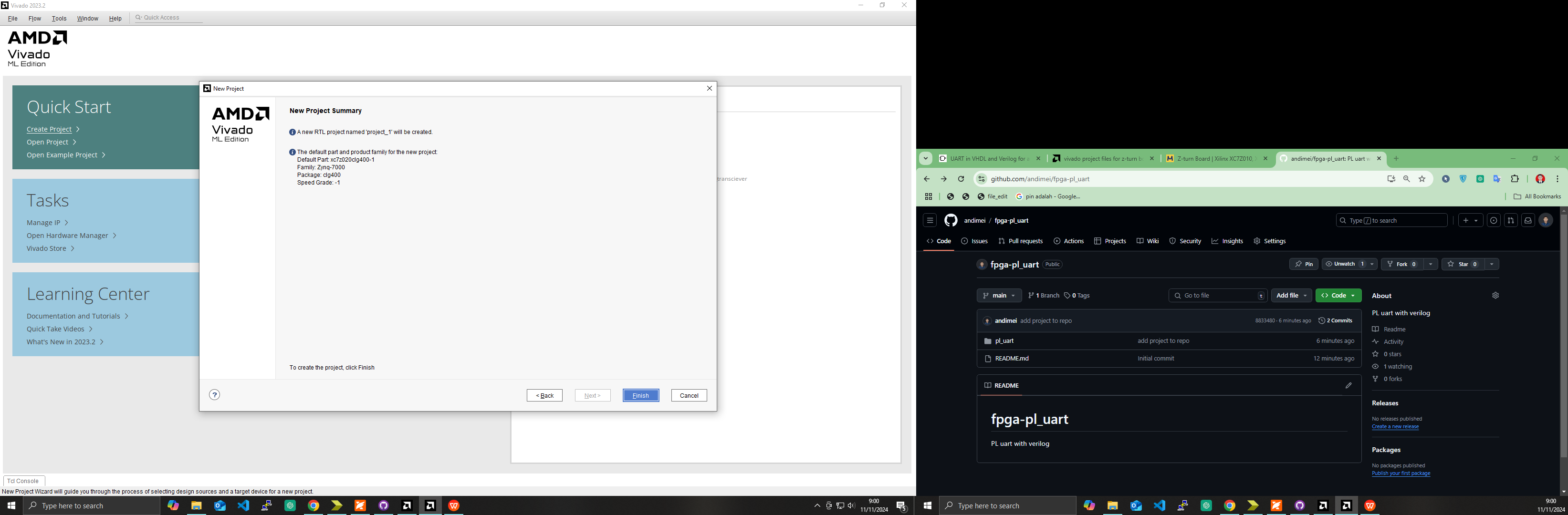
1. Buat project baru

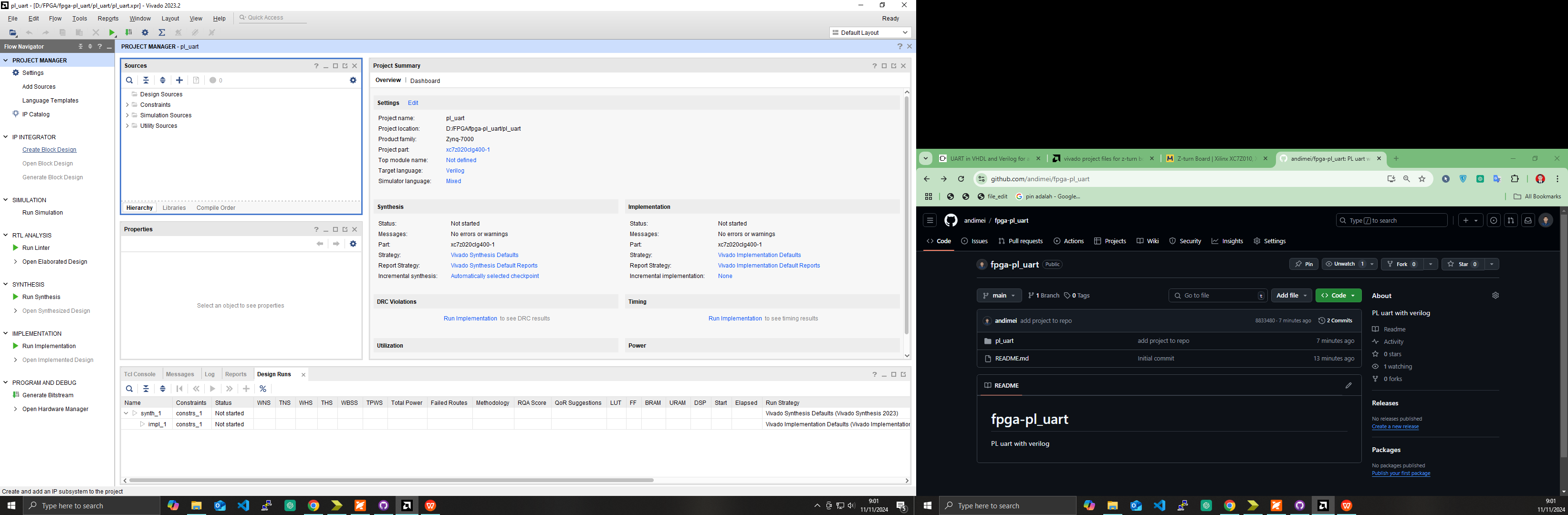


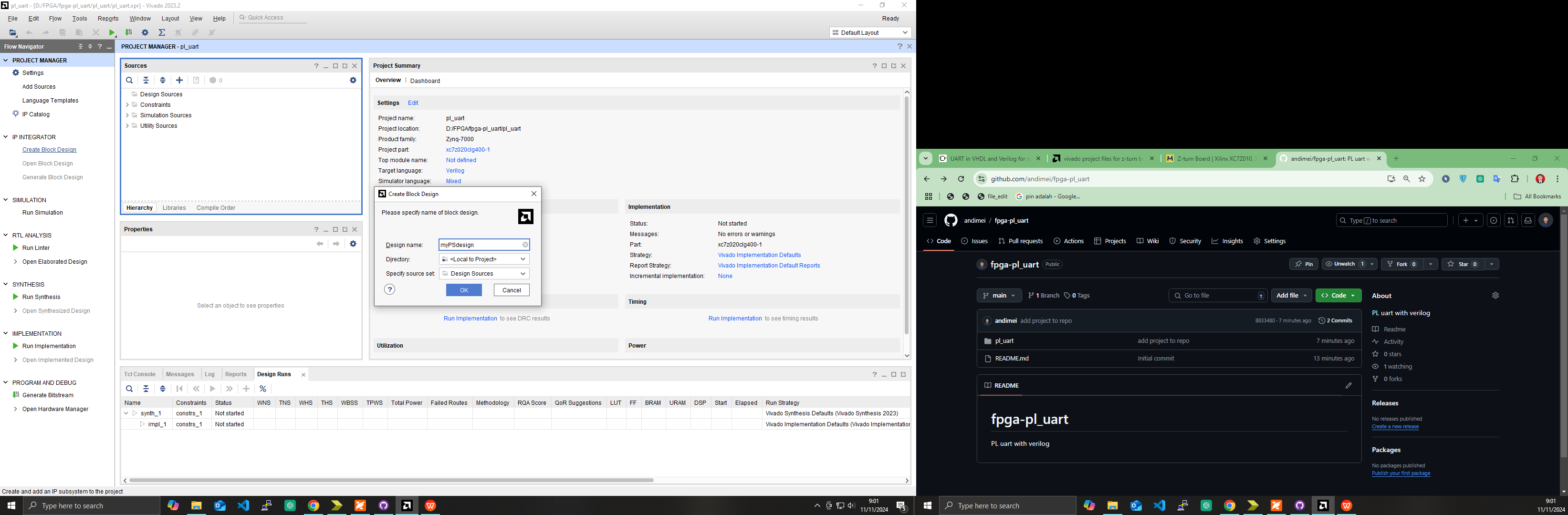


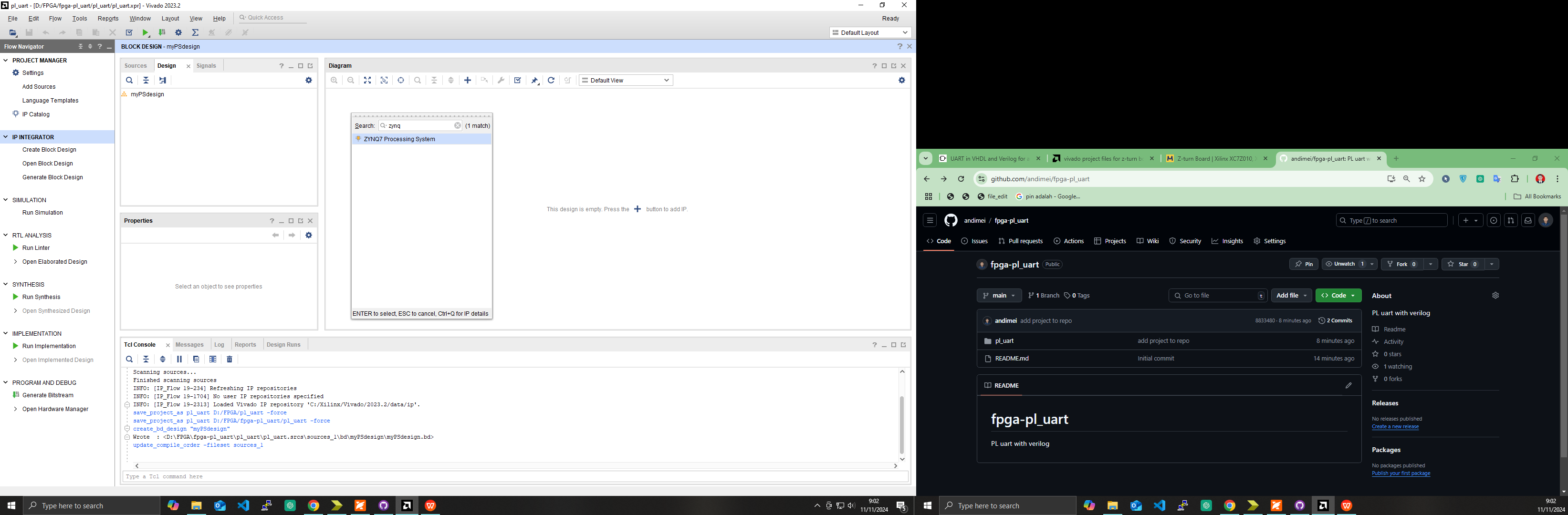


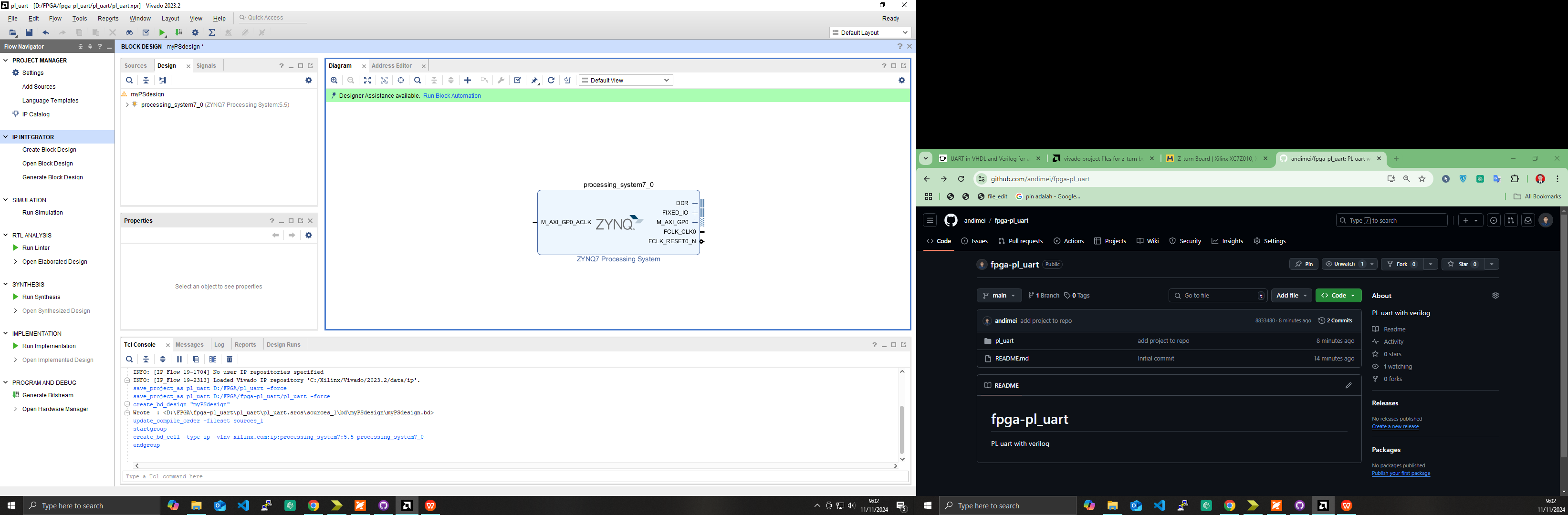


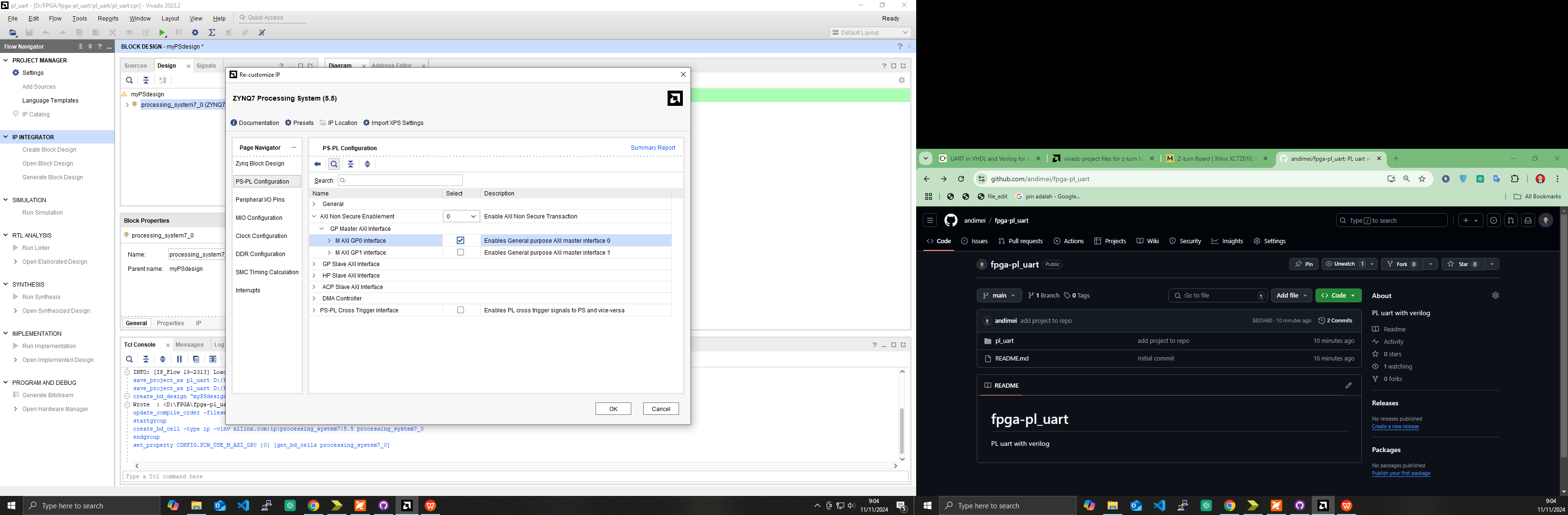


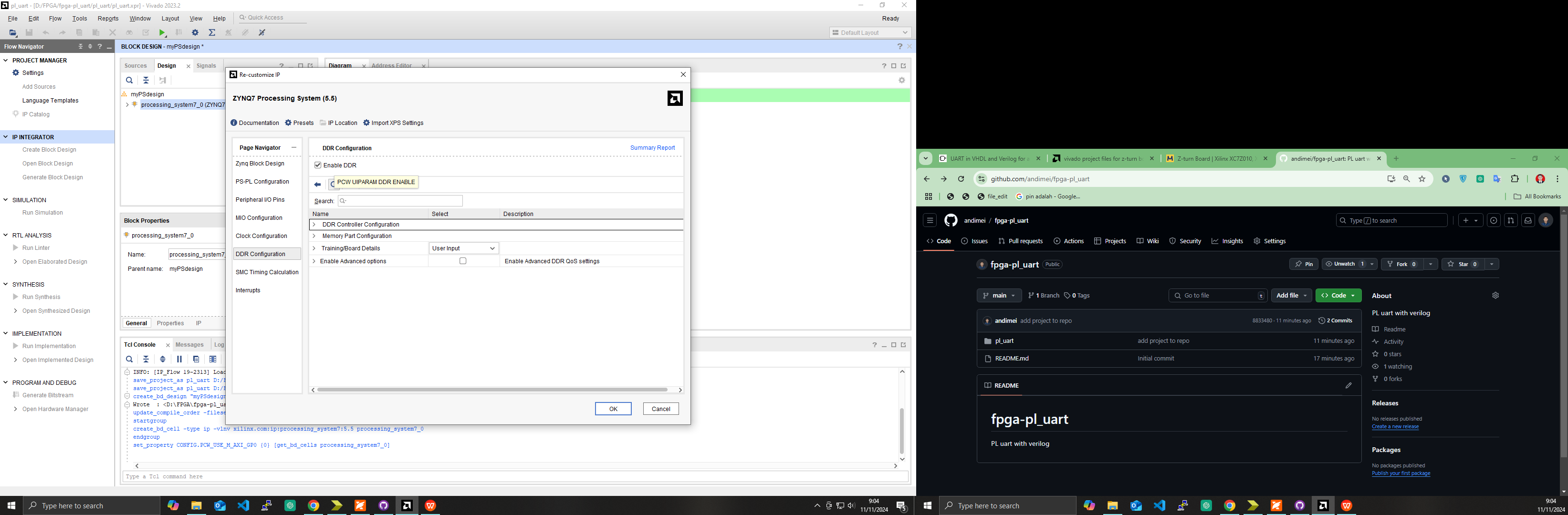


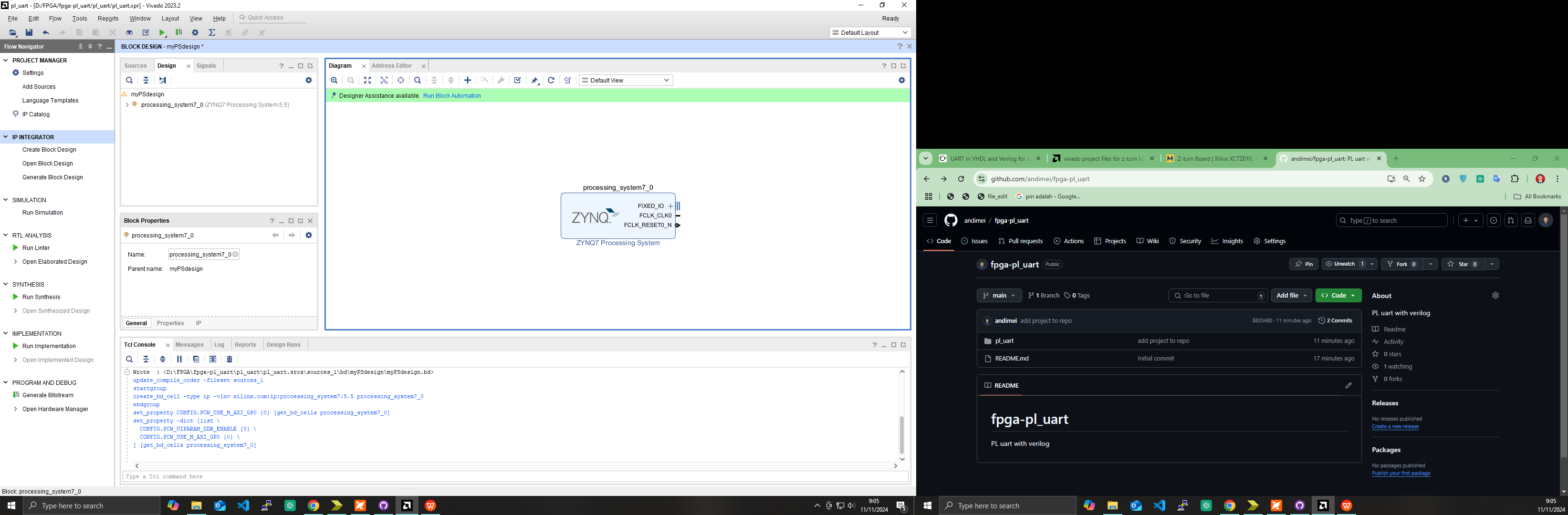


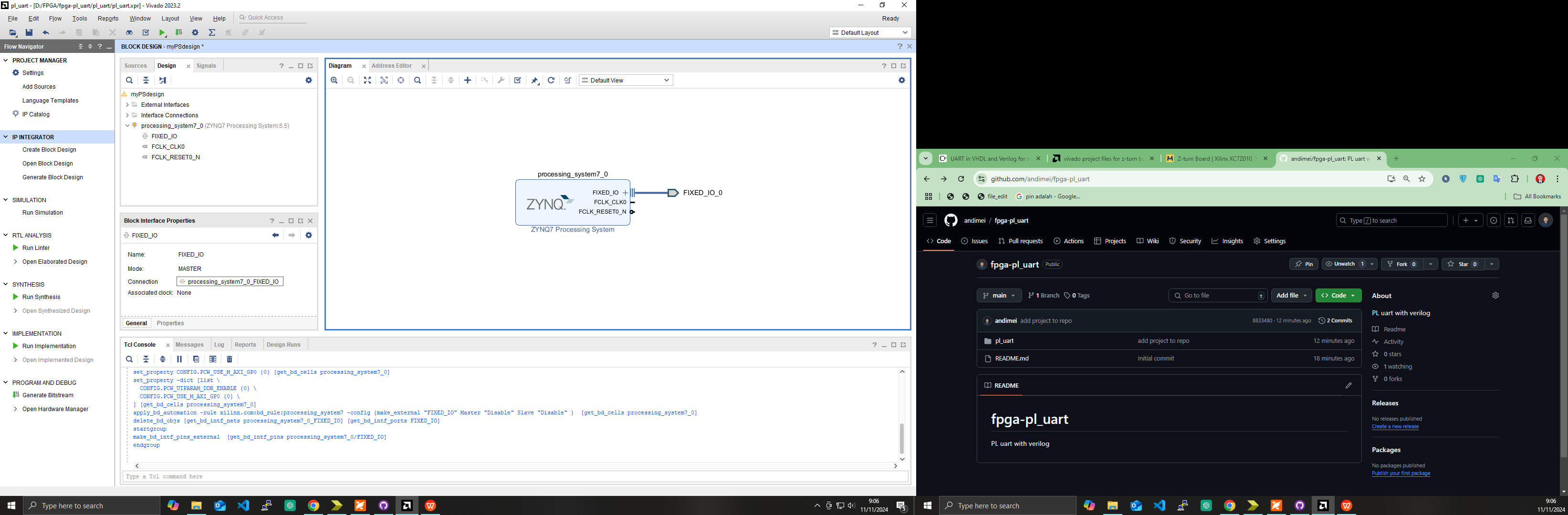


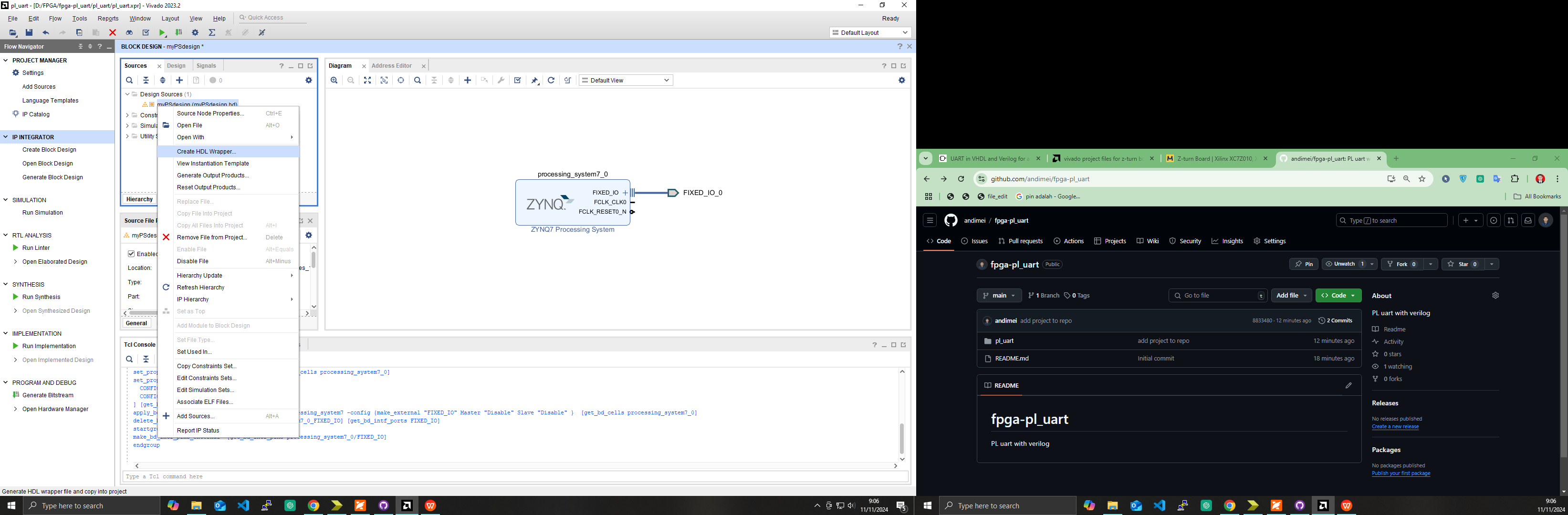


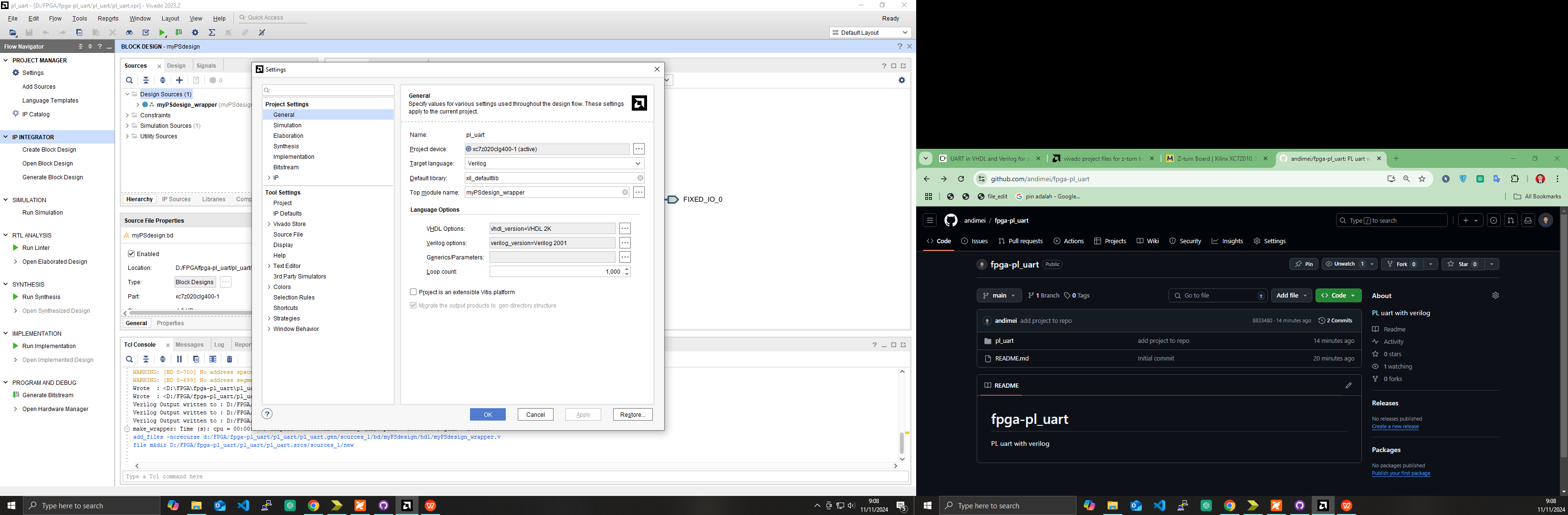


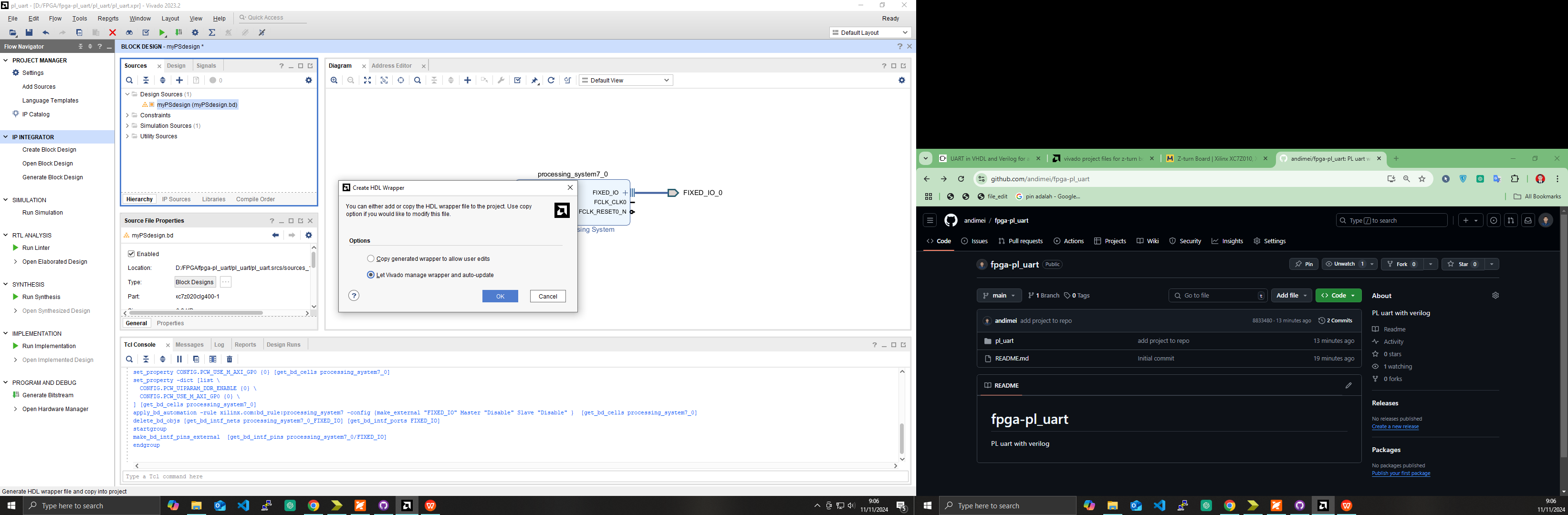


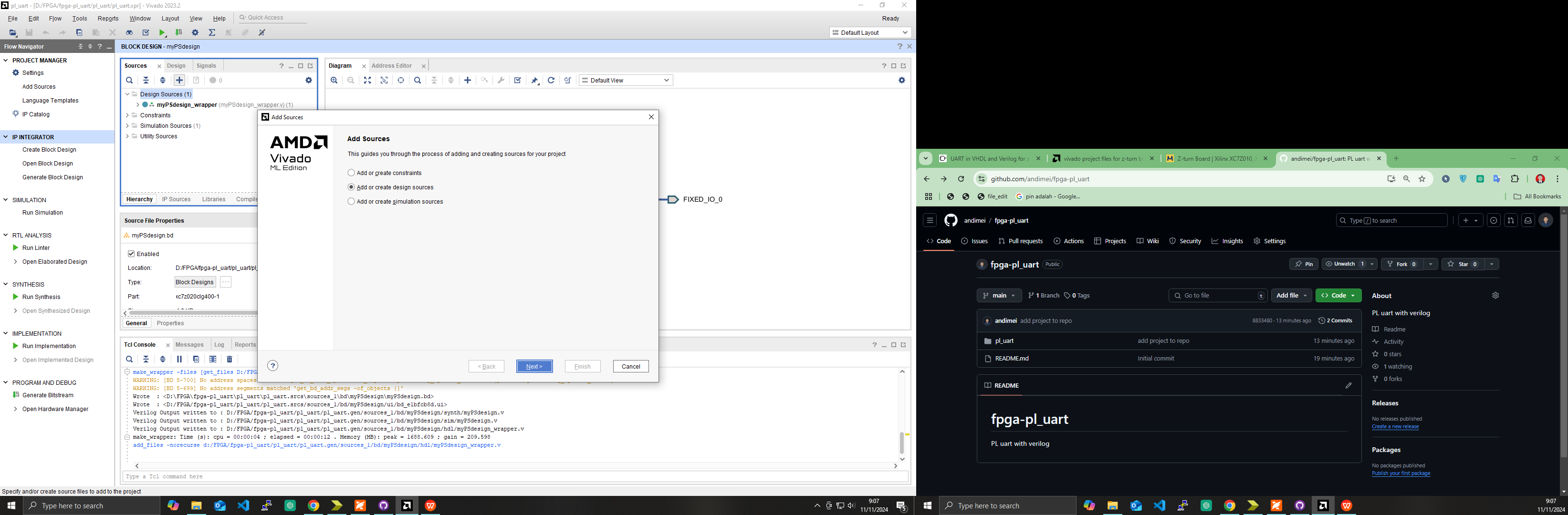


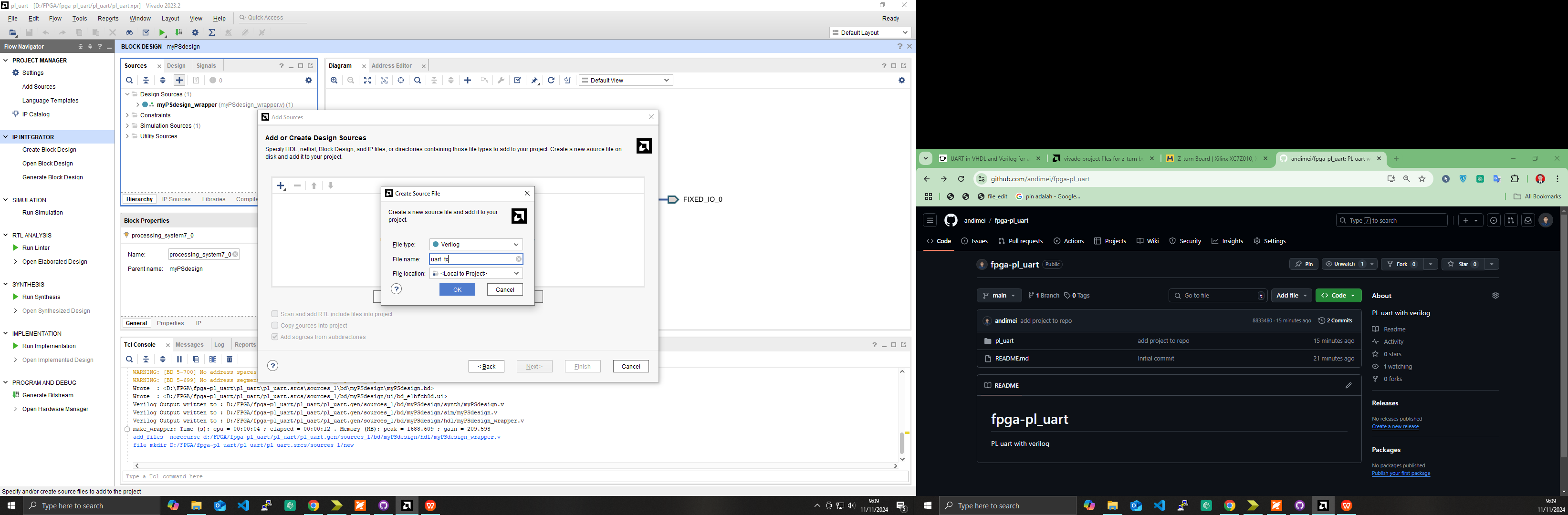


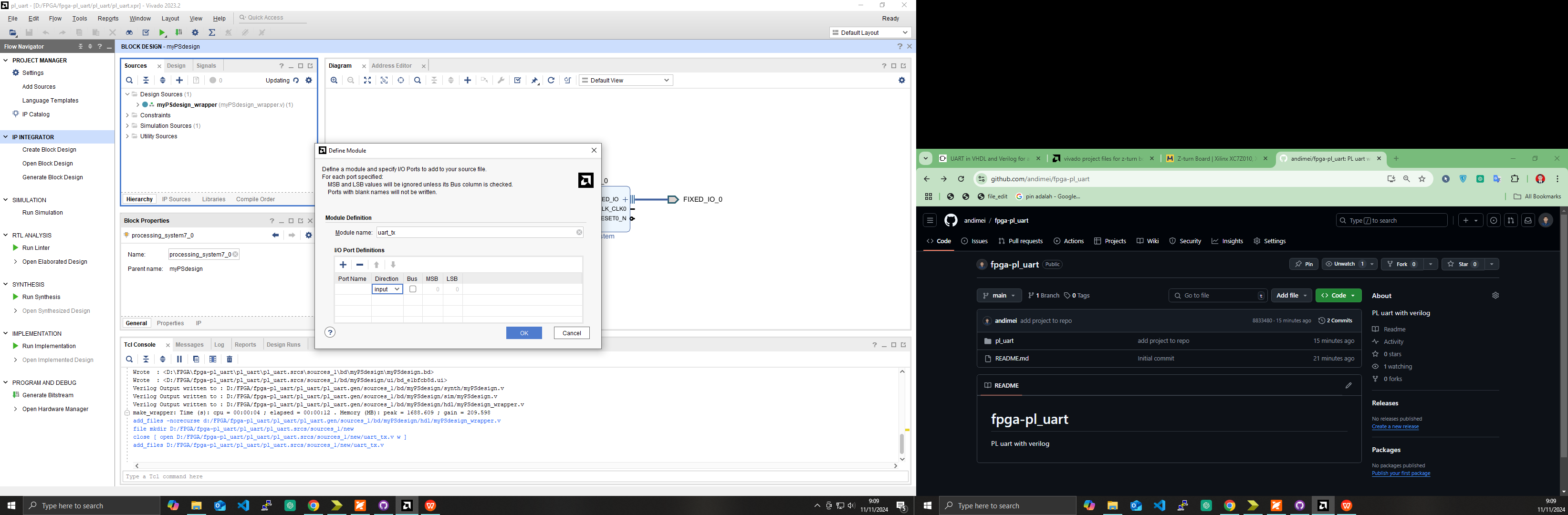


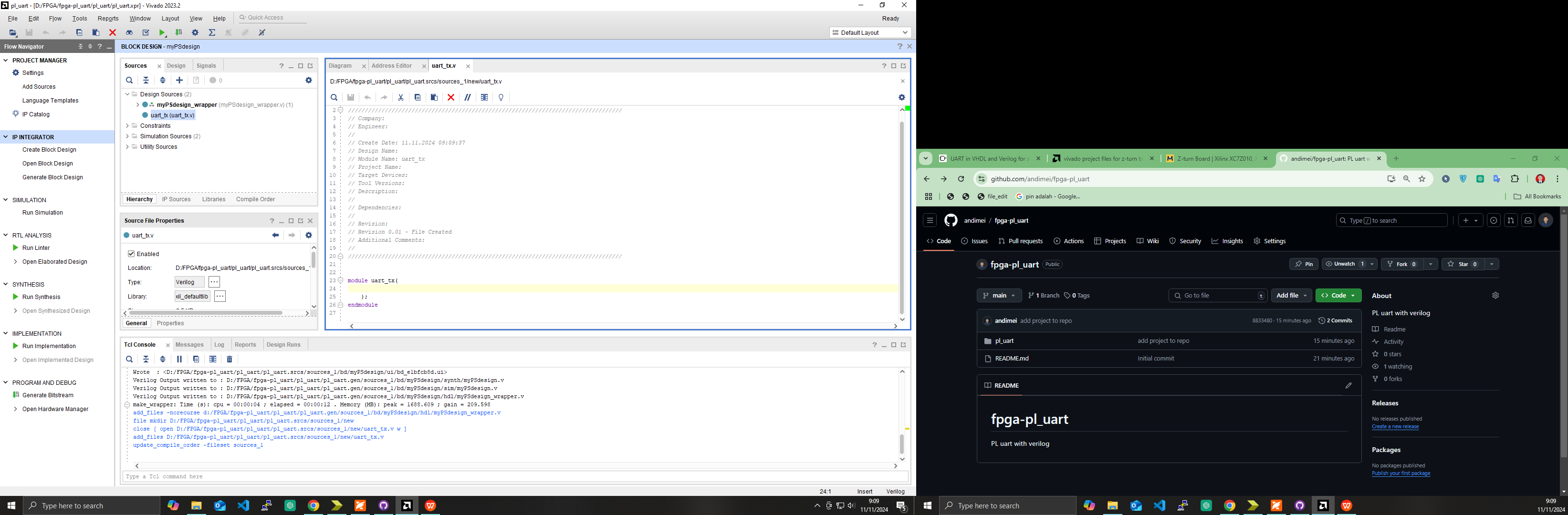


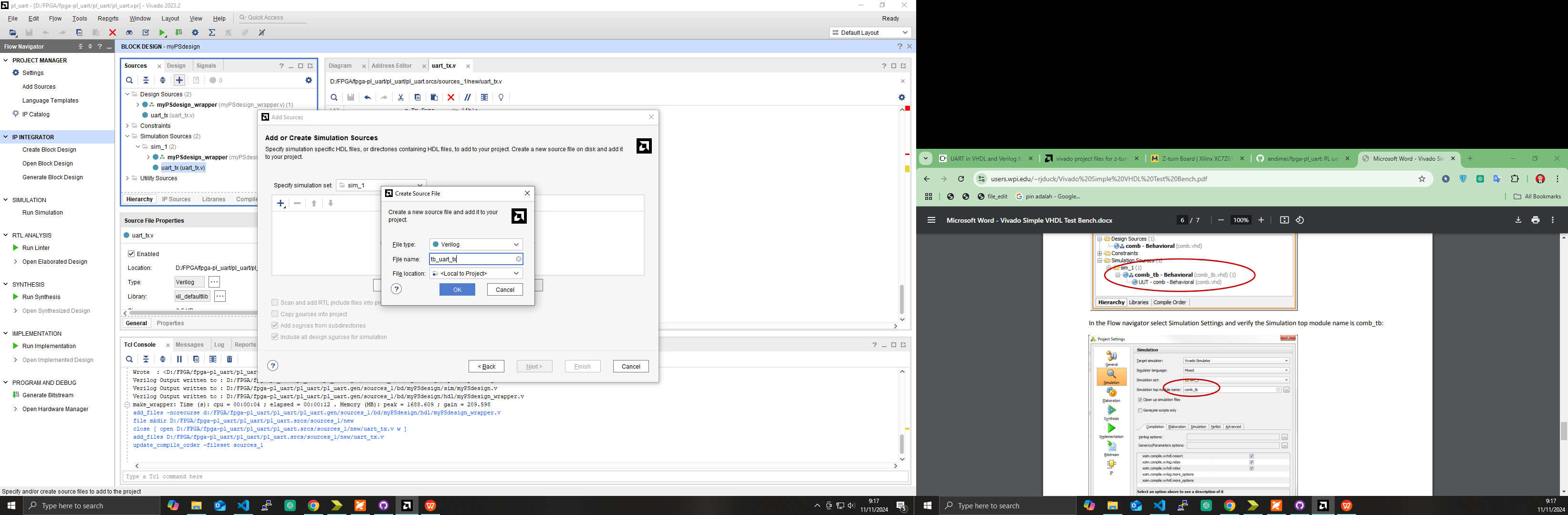












· c\_ : Menandakan **constant** atau konstanta yang nilainya tetap dan tidak berubah selama simulasi. Biasanya berupa parameter atau nilai tetap dalam testbench.

· r\_ : Menandakan **register** atau sinyal yang disimpan dengan flip-flop. Dalam Verilog, ini sering digunakan untuk variabel yang diassign dalam blok always.

· s\_ : Menandakan **signal** atau sinyal biasa, yang bisa berupa wire atau reg tetapi biasanya hanya digunakan untuk komunikasi antarblok atau komponen.  
Berikut penjelasan mengenai arti masing-masing huruf kecil tersebut:

**s** pada s\_next dan s\_reg:

* 1. s biasanya singkatan dari **"state"** atau **"sub-tick"**.
  2. Dalam konteks ini, s\_reg dan s\_next digunakan untuk menghitung jumlah sub-tick atau siklus dalam satu bit (sering kali digunakan dalam komunikasi serial untuk mencapai frekuensi bit yang tepat).
  3. Ini bertugas melacak penghitungan di antara sinyal clock (misalnya, sampai 16 tick sebelum melanjutkan ke tahap berikutnya).

**n** pada n\_next dan n\_reg:

* 1. n sering digunakan sebagai singkatan untuk **"bit number"** atau **"index"**.
  2. Di sini, n\_reg dan n\_next menghitung posisi bit saat mengirim data byte per bit (dalam contoh ini, byte dengan 8 bit), yang menunjukkan berapa banyak bit yang telah dikirim.

**b** pada b\_next dan b\_reg:

* 1. b umumnya singkatan dari **"byte"** atau **"buffer"**.
  2. b\_reg dan b\_next menyimpan data byte yang akan dikirim bit demi bit. Pada setiap siklus, bit paling rendah dari b\_reg dikirim melalui output (tx\_next = b\_reg[0]), dan kemudian b\_reg di-shift untuk menyiapkan bit berikutnya untuk transmisi.

### Kesimpulan

Konvensi penamaan ini membantu menunjukkan tujuan variabel:

* **s**: untuk penghitungan siklus atau sub-tick,
* **n**: untuk menghitung indeks bit dalam byte,
* **b**: untuk menyimpan byte data yang sedang ditransmisikan.

Dengan mengikuti konvensi ini, kode menjadi lebih jelas dan lebih mudah dipahami tanpa perlu komentar tambahan yang banyak.

Divisor=Baud RateClock Frequency​=115,20050,000,000​≈434.027