DynaRapid: Fast-Tracking From C to Routed Circuits

Andrea Guerrieri, Srijeet Guha, Chris Lavin, Eddie Hung, Lana Josipović, and Paolo Ienne
andrea.guerrieri@ieee.org

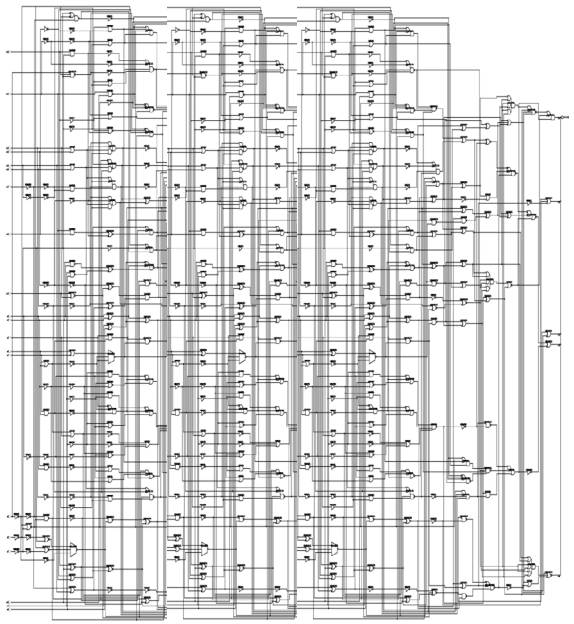
FPL Conference 2024
Sept 02-06, 2024, Turin, Italy

From C to ~~RTL~~ P&R Circuits

```
x = A[0];
y = B[0];
i = 1;
do {
   if(cond) {
      A[i] = x;
   } else {
      B[i] = y;
   }
   i++;
} while(i < size);
z = x + y;
```

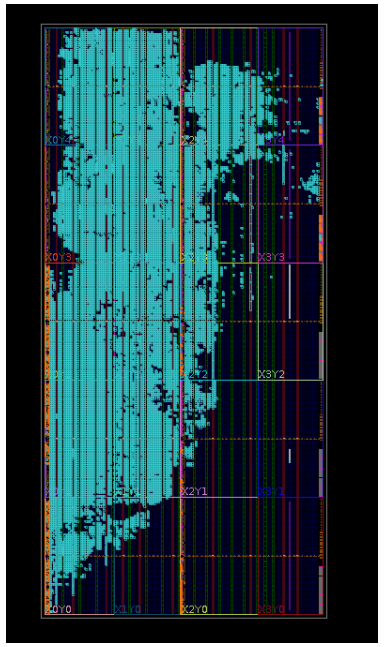
Software

HLS



Circuit (RTL)

LS + P&R



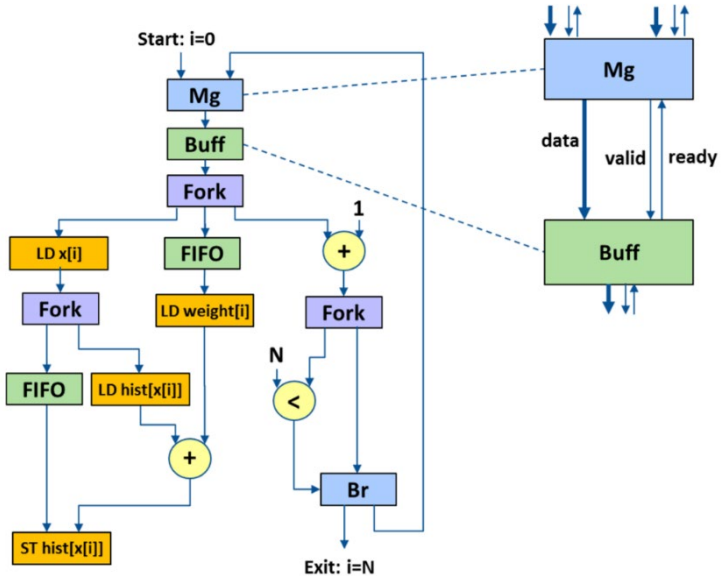
Placed and routed

Huge time gap

Speed up design flow from C to placed and routed circuits from minutes to seconds

Dynamatic: Open-Source HLS Compiler

Dynamatic generates dynamically scheduled circuits out of C code



- The circuits are exclusively composed of **a limited set of elastic components**

https://dynamatic.epfl.ch/

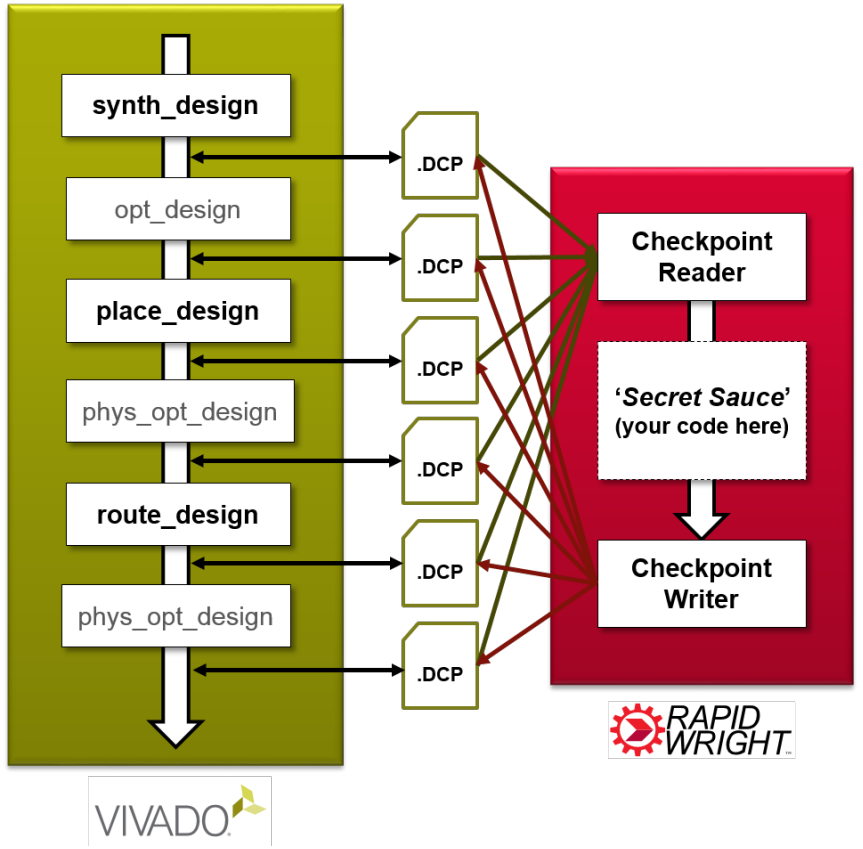
Josipović, Guerrieri, and Ienne. Dynamatic: From C/C++ to Dynamically-Scheduled Circuits. Invited tutorial. FPGA 2020

RapidWright: Enabling Custom Crafted Implementations for FPGAs

An open-source platform with a gateway to backend tools in Vivado™

RapidWright provides APIs enabling users to

- customize implementations
- manipulate and relocate pre-implemented designs



https://www.rapidwright.io/

Lavin and Kaviani, "RapidWright: Enabling custom crafted implementations for FPGAs," FCCM 2018

From C to ~~RTL~~ P&R Circuits

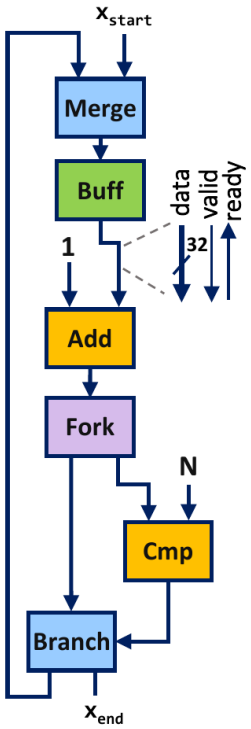
Speed up the design flow from several minutes to seconds

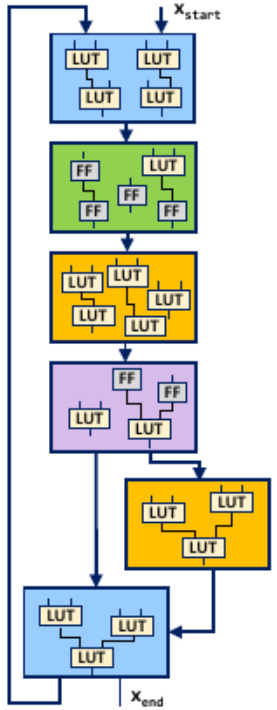
HOW??

```
x = A[0];
y = B[0];
i = 1;
do {
   if(cond) {
      A[i] = x;
   } else {
      B[i] = y;
   }
   i++;
} while(i < size);
z = x + y;
```

Software

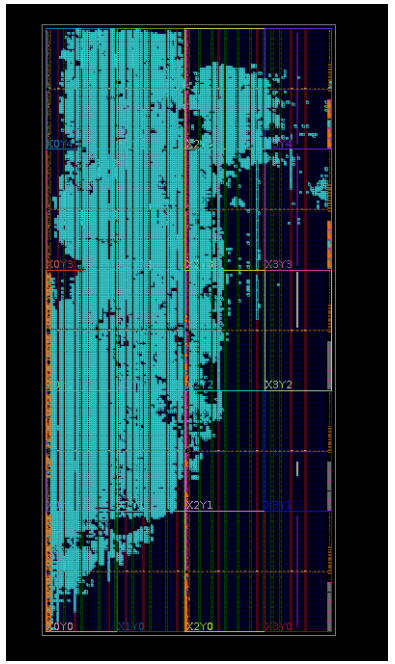
HLS





Circuit (RTL)

LS+P&R

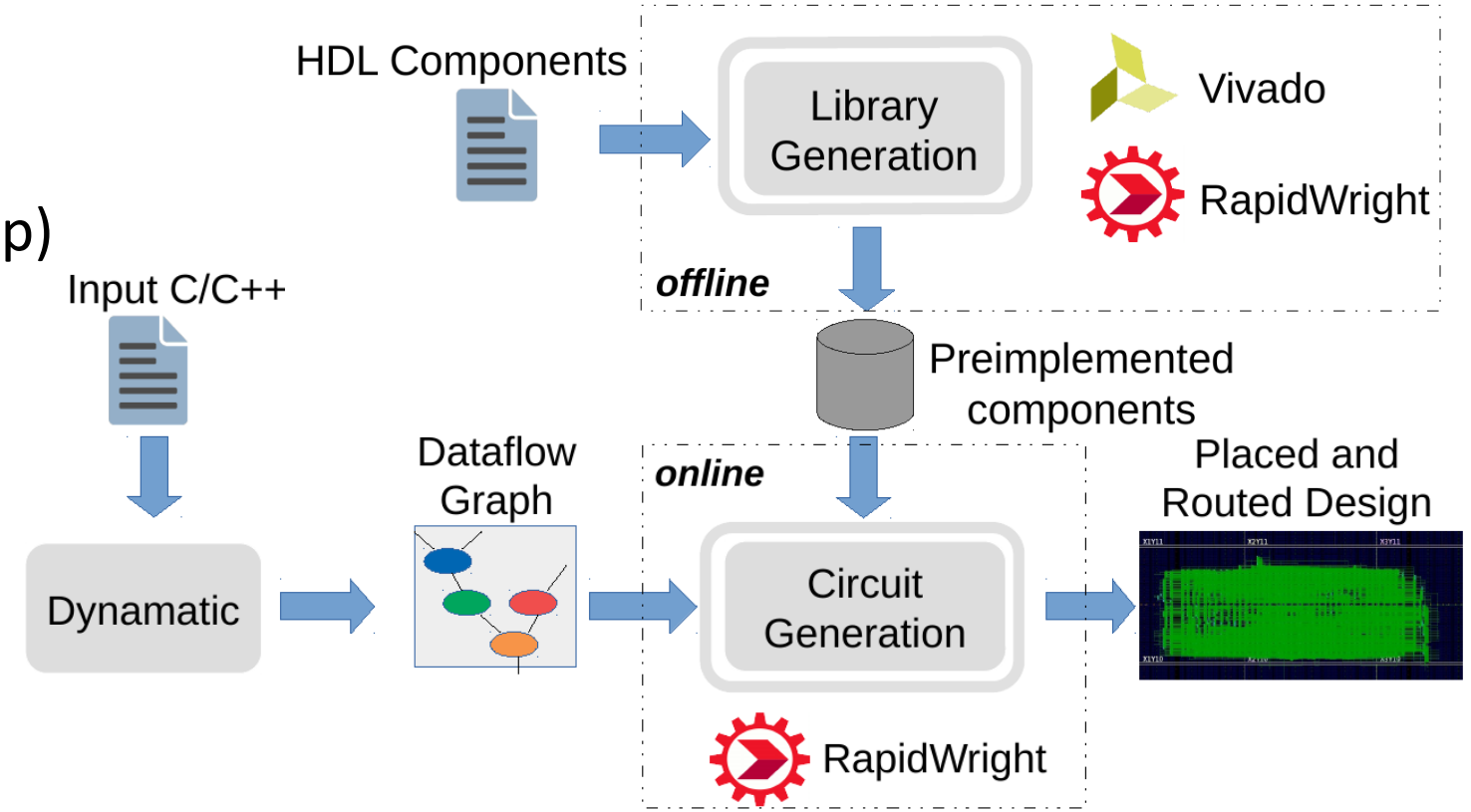


Placed and routed

Exploiting structure to reduce complexity

DynaRapid Design Flow

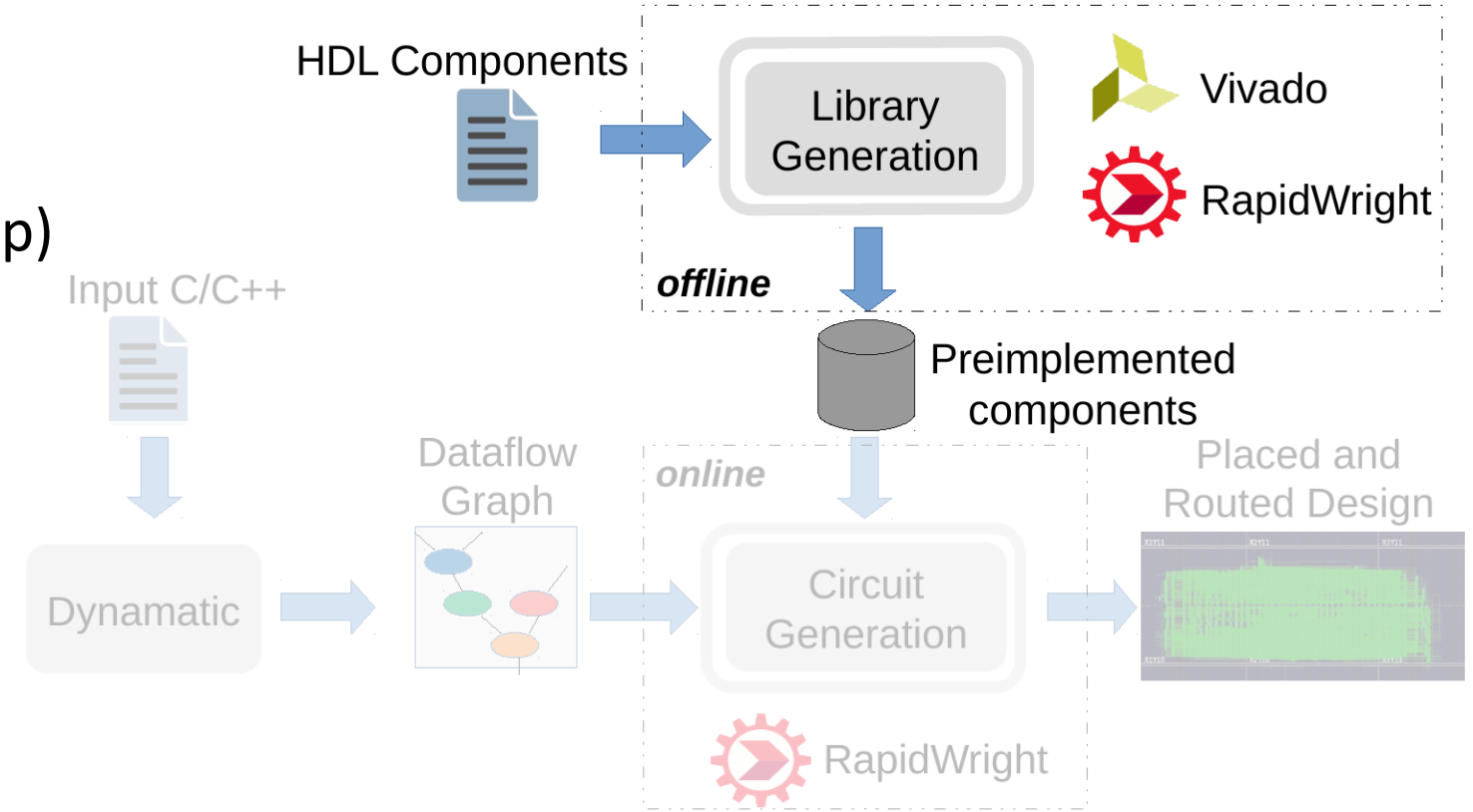
- **Library generation:** performed offline, once per FPGA device (e.g., Virtex UltraScale+™ xcvu13p)
- **Circuit generation:** performed online (load modules, stitch, place, and route)



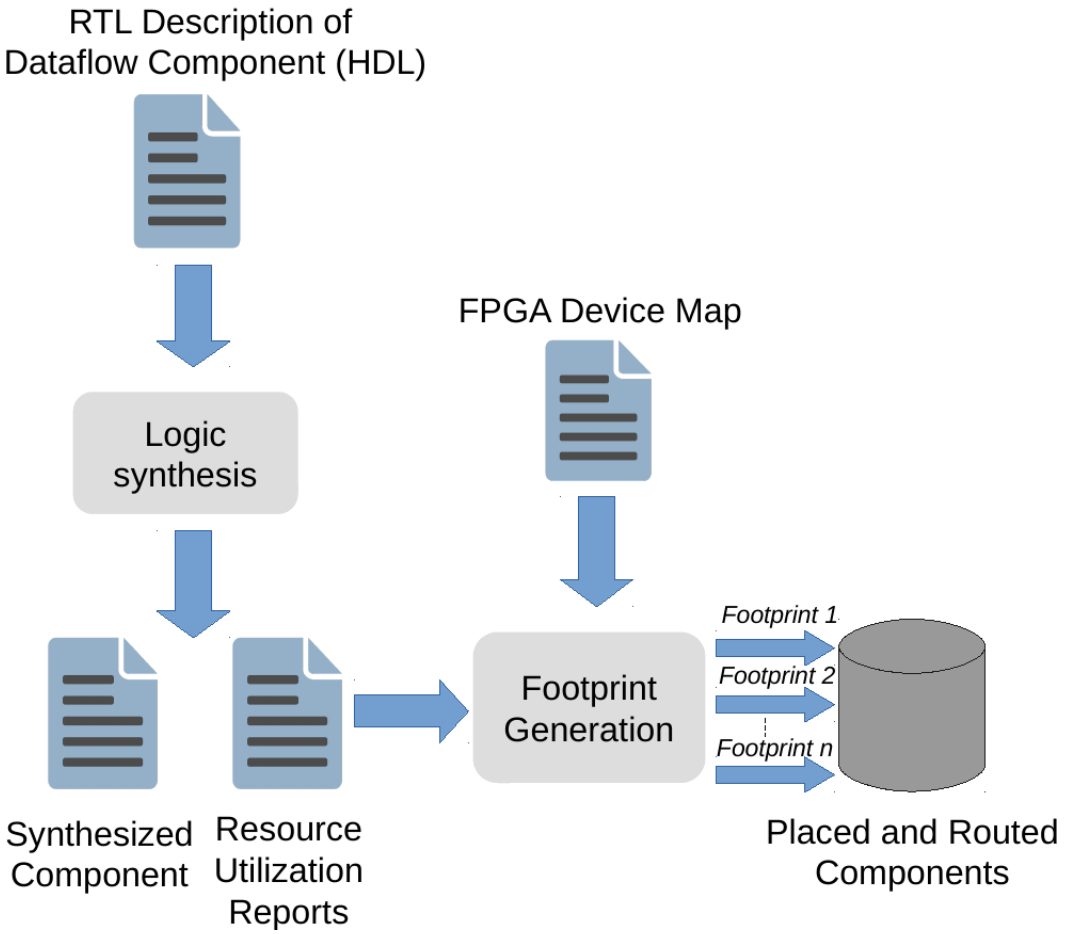
The output is a fully placed-and-routed design checkpoint

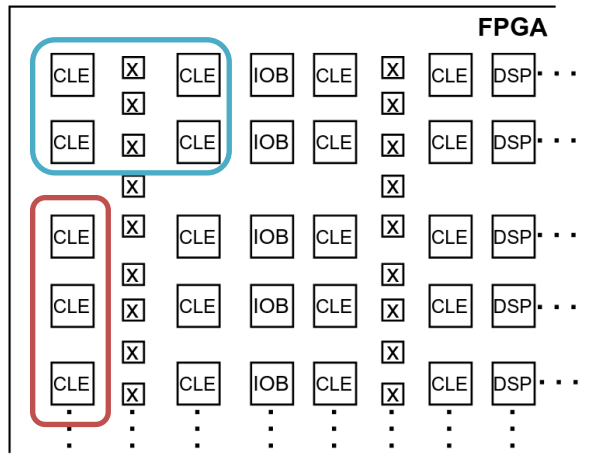
DynaRapid Design Flow

- **Library generation:** performed offline, once per FPGA device (e.g., Virtex UltraScale+™ xcvu13p)
- **Circuit generation:** performed online (load modules, stitch, place, and route)



Library Generation



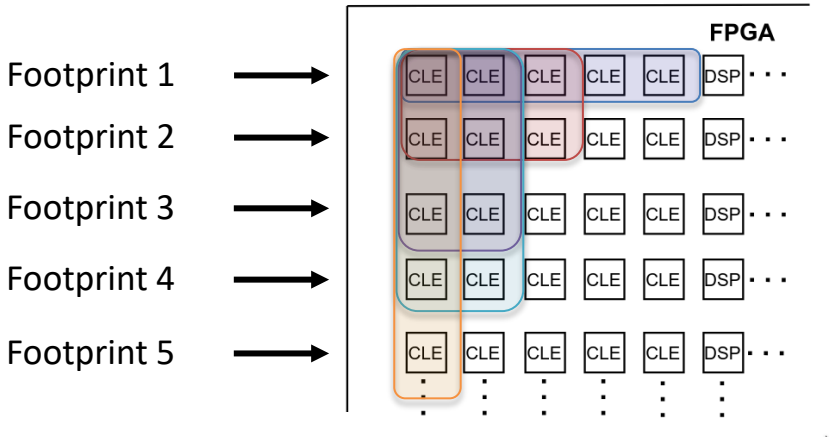


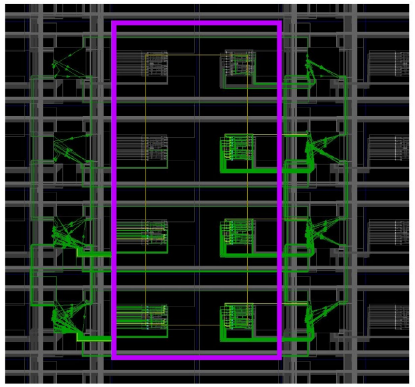
Create efficient legal footprints

- **minimizing** occupancy of physical resources
- **preserving** routability
  - intra-module (internal connections)
  - inter-module (I/O connections)

Minimizing Occupancy

We explore rectangular footprints of increasing height that are wide enough to accommodate the fully routed component





Example of footprint

Technical Challenges

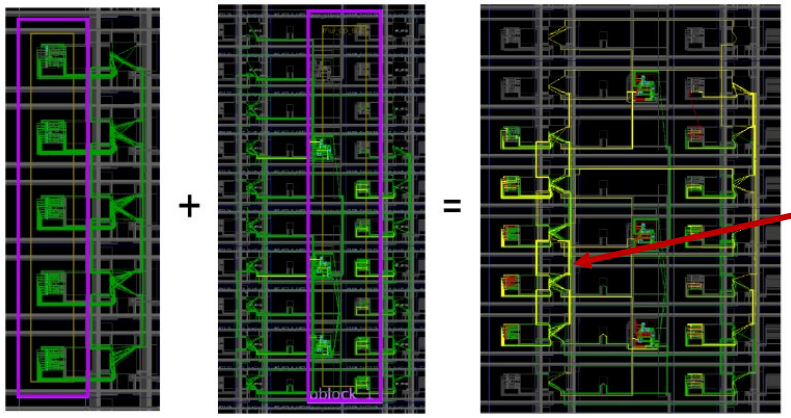
No direct control of routing resources (switch boxes) while drawing footprint

- Module placement: resource overlapping
- Module routing: I/O reachability

Module Placement: Resource Overlapping

Different elastic modules need to be placed together on the same floorplan

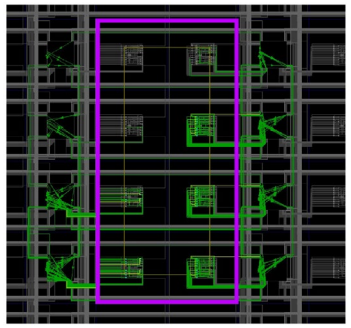
- Routing resources might overlap while placed adjacently



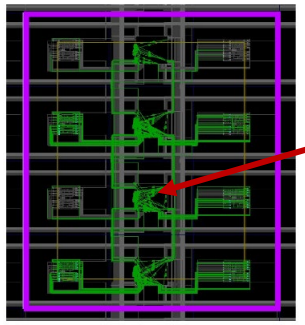
FIFO Multiplier Combined design

The yellow lines are disconnected wires (antennas) and the implementation is invalid

- Methods to retain footprints containing both logic and routing resources

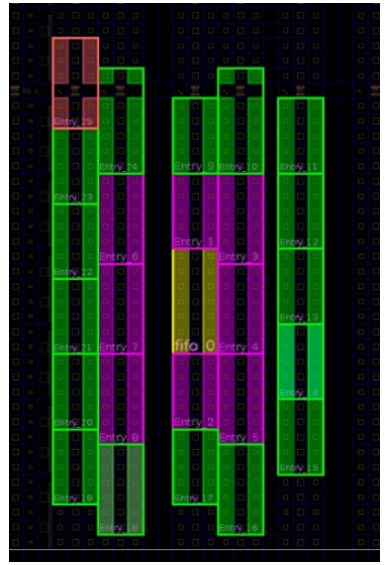


Bad footprint



Good footprint

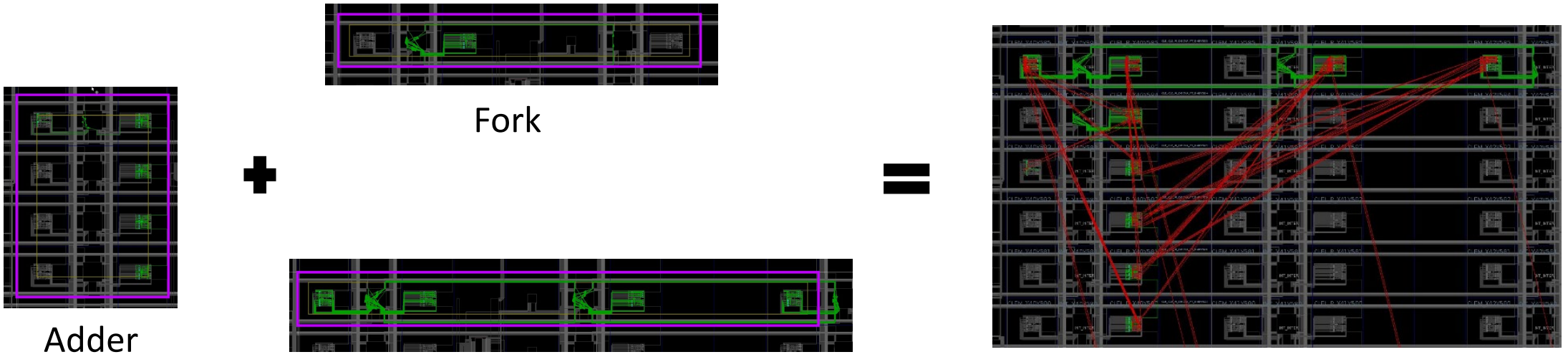
The green lines are inside the footprint



Module Routing: I/O Reachability

Different elastic modules need to be routed together on the same floorplan

- No guarantee that the module's external I/O pins enough routing resources

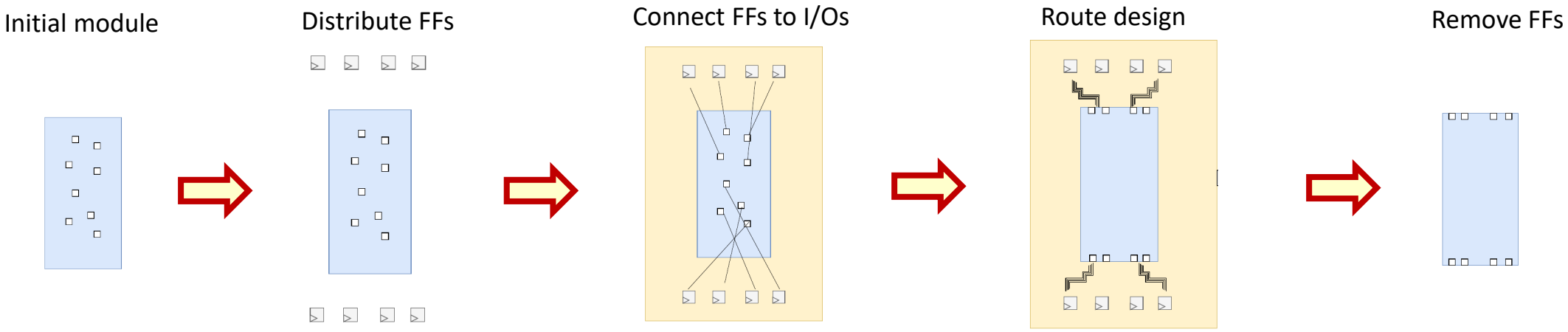


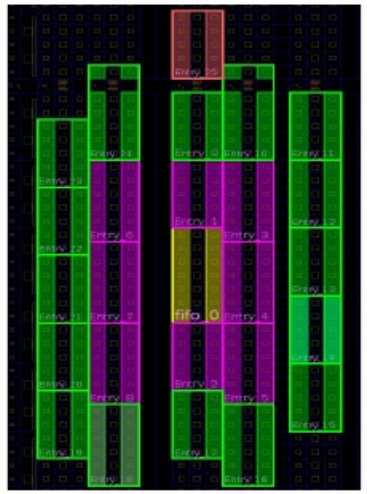
Combined design. The red lines are routing errors

- 32-bit fork with one input and six outputs has $(1 + 6) \cdot (32 + 2) = 238$ I/Os

Module Routing: I/O Reachability

Post-processing technique to expose module's I/Os

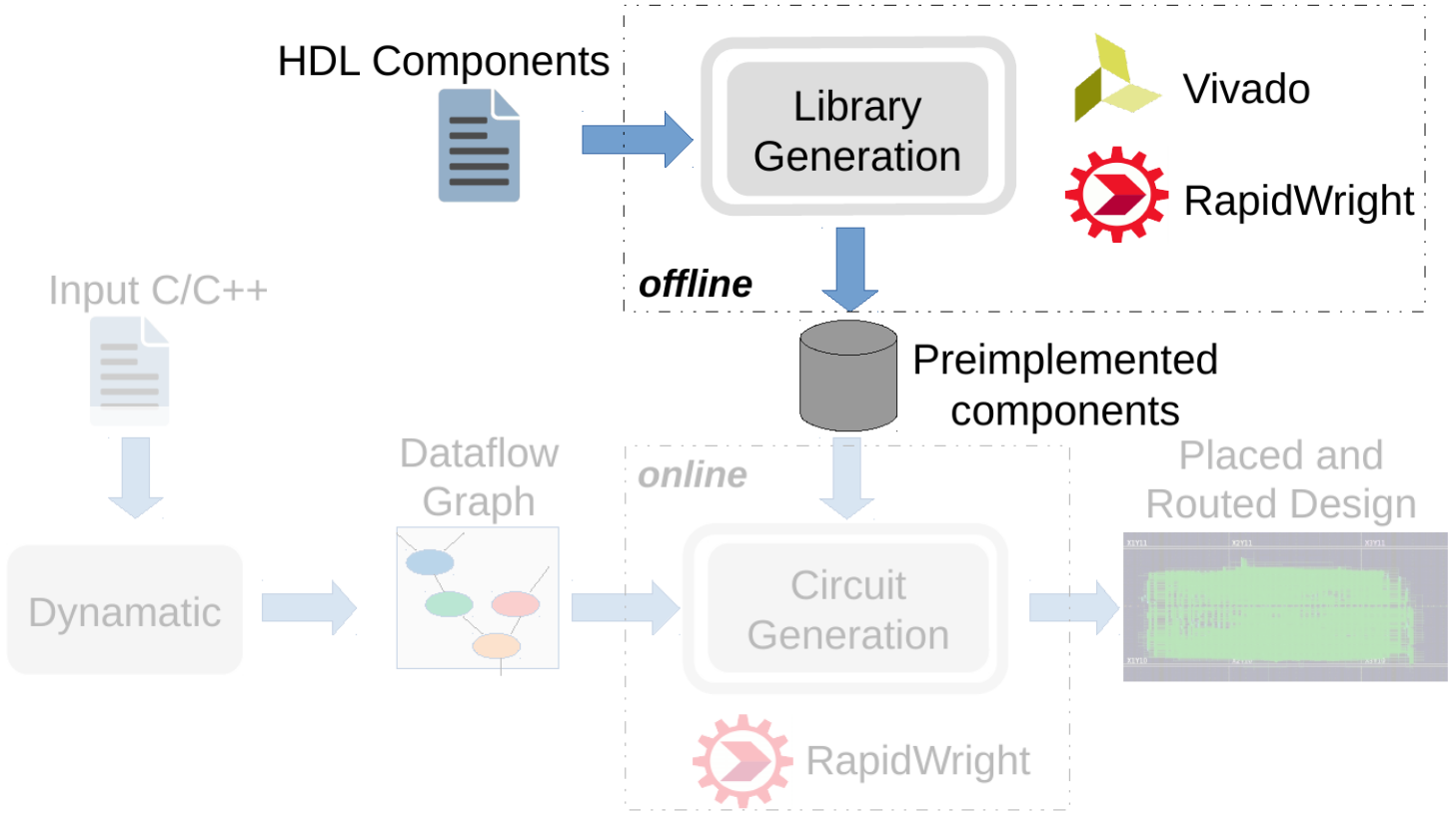




Modules can be assembled on the same floorplan and fully routed with no routing errors

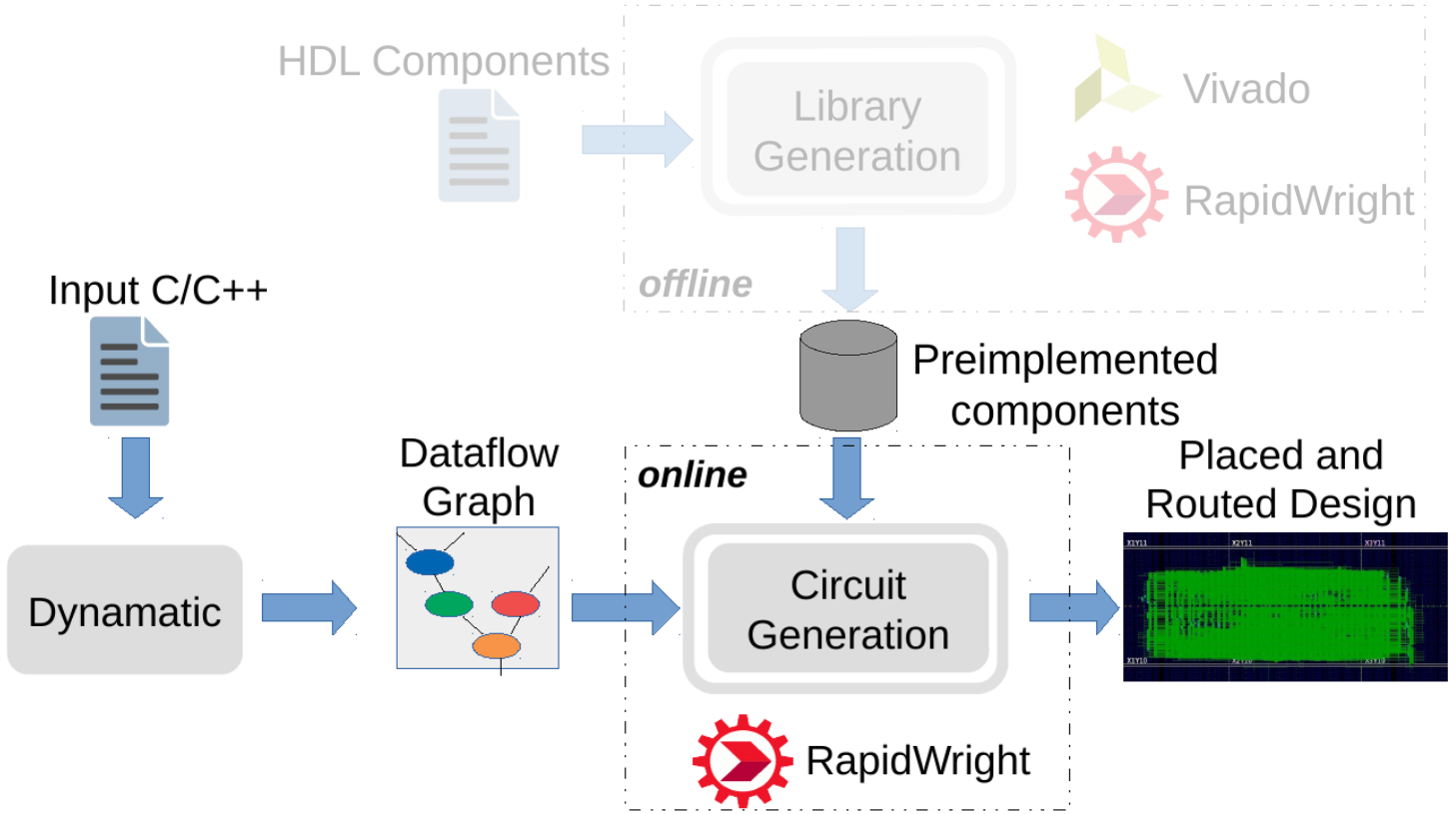
DynaRapid Design Flow

- **Library generation:** performed offline, once per FPGA device
  - 74 elastic components (32-bit)
  - up to 50 footprints (Ultrascale+™)
  - up to 60k locations (xcvu13p)
- **Circuit generation:** performed online (load modules, stitch, place, and route)



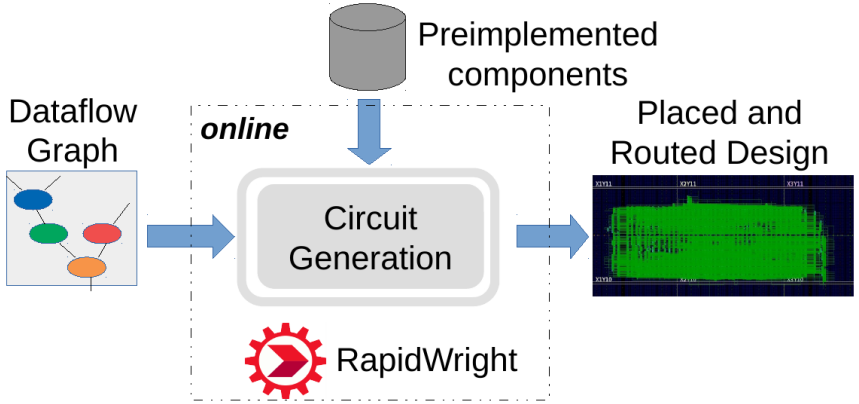
DynaRapid Design Flow

- **Library generation:** performed offline, once per FPGA device
  - 74 elastic components (32-bit)
  - up to 50 footprints (Ultrascale+™)
  - up to 60k locations (xcvu13p)
- **Circuit generation:** performed online (load modules, stitch, place, and route)



Circuit Generation

The circuit generation is performed online: load modules, stitch, place, and route



Runtime first!

Placement

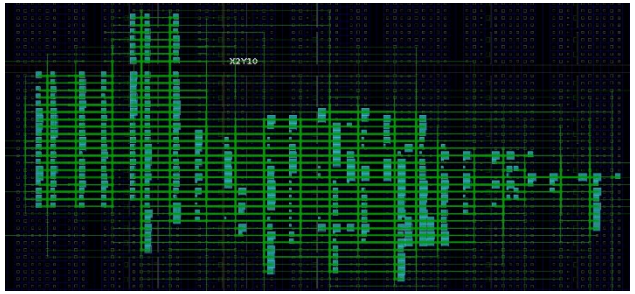
Very fast placement strategy to find legal solution

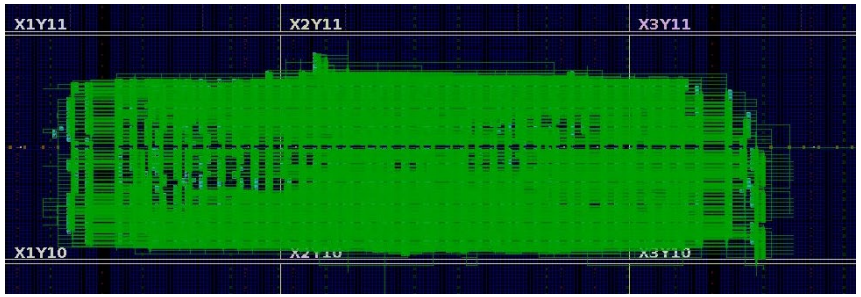
- selection of the best footprint configuration and best location

Routing

Non-timing-driven techniques

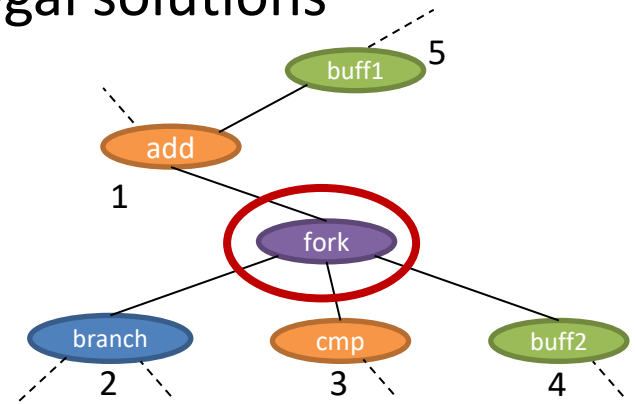
- allowing partial routing





Placement

We developed a simple but effective greedy placer to quickly find legal solutions

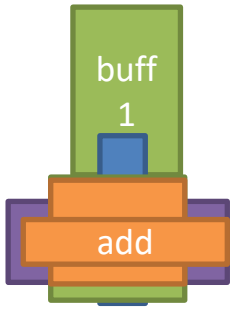


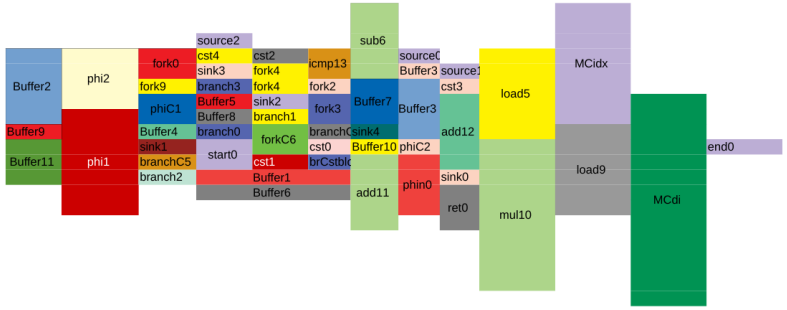
Placement order

(1) Identify the “root” component in the graph with more connection
(2) Set the module’s placement order using breadth-first traversal

Placement location

(1) Define the position of the root
(2) Explore for suitable nearby positions circularly
(3) Consider all available footprints and we select the one with minimum Manhattan distance



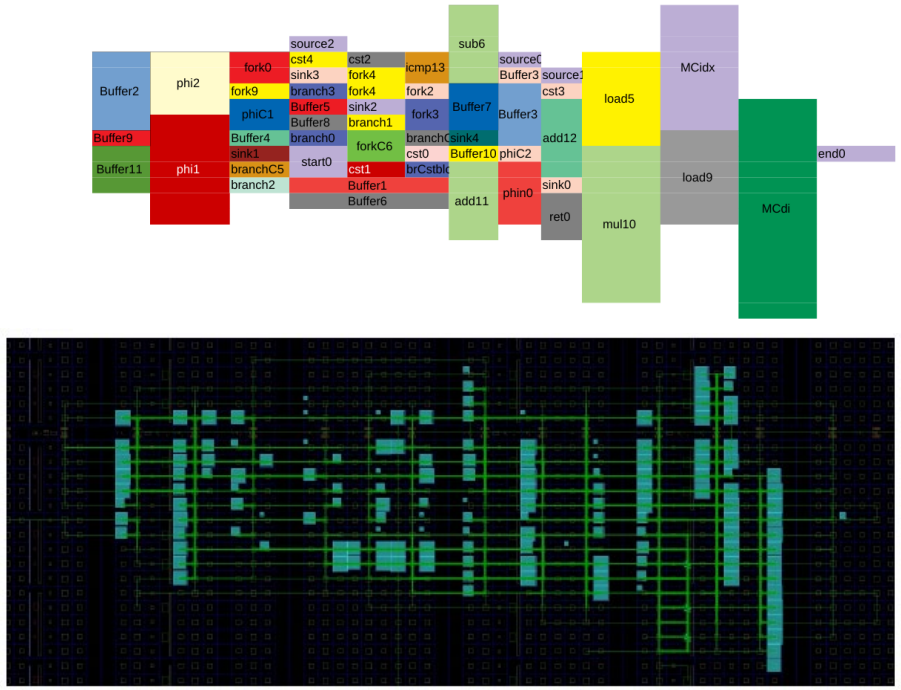


Greedy Placement Result. The circuit is a Finite Impulse Response filter and each color corresponds to a different elastic component.

Routing

The elastic modules are stitched and need to be routed to generate a valid circuit

- We use a modified version of RWRoute (partial router) in non-timing-driven mode



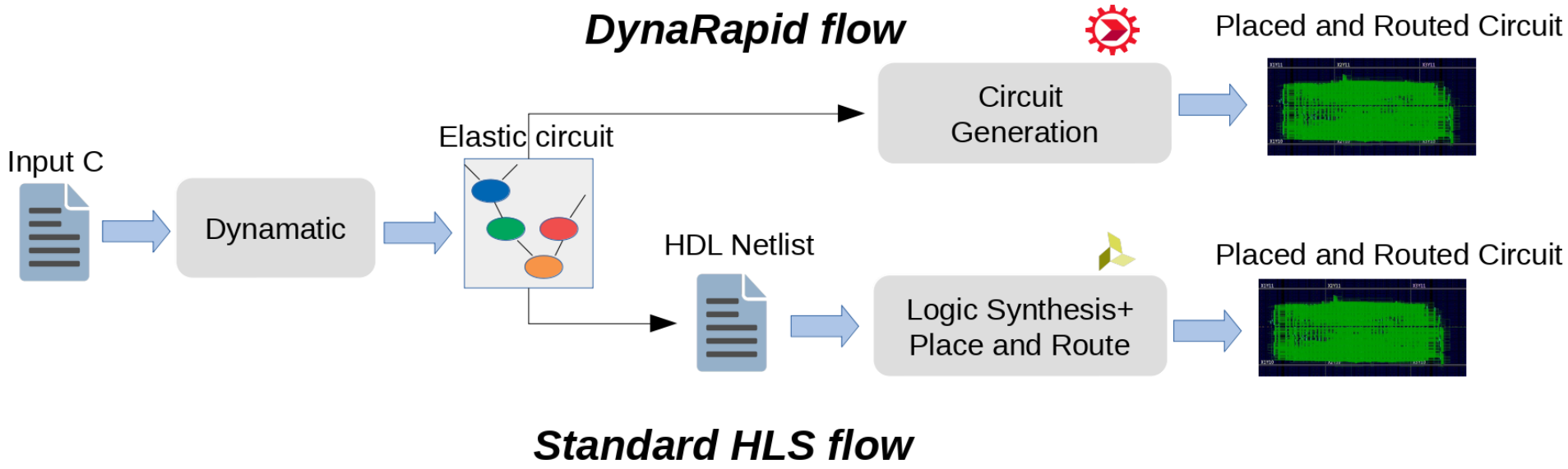
Final placed and routed design.
The circuit is a Finite Impulse Response filter.

We generate fully routed designs with no routing errors

Zhou, Maidee, Lavin, Kaviani, and Stroobandt, “RWRoute: An open-source timing-driven router for commercial FPGAs”, ACM TRETS, Nov. 2021.

Evaluation

We compared DynaRapid with Vivado™ 2023.1, targeting a Virtex UltraScale+™ xcvu13p



We configure Vivado to achieve the fastest implementation:

- **Flow Runtime Optimized strategies** (Logic synthesis)
- **-directive Quick**, fastest non-timing-driven compile time (P&R)

Vivado Design Suite User Guide: Synthesis, AMD Inc., 2023. [Online]. Available: https://docs.xilinx.com/r/en-US/ug901-vivado-synthesis/Vivado-Preconfigured-Strategies

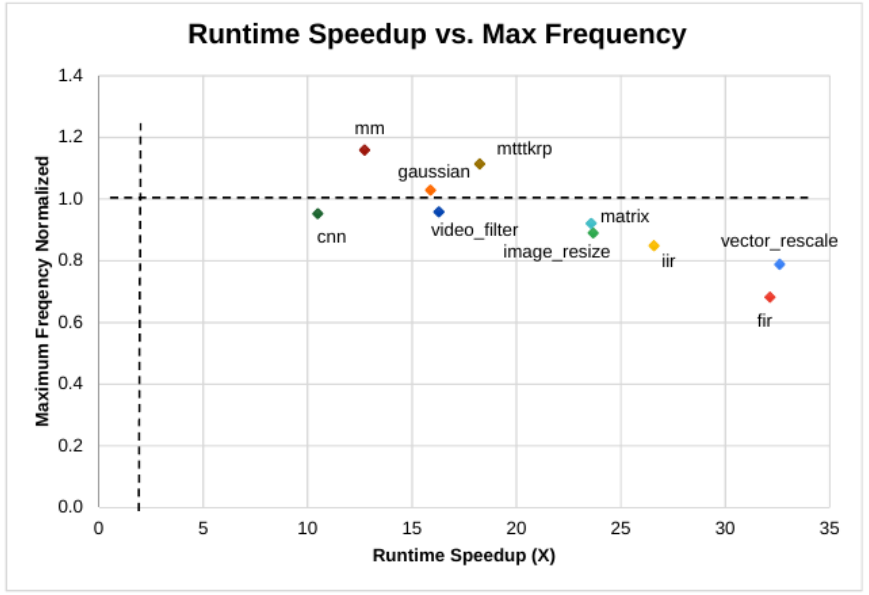
Vivado Design Suite, Xilinx Inc., 2023. [Online]. Available: http://www.xilinx.com/products/design-tools/vivado.html

Experimental Results

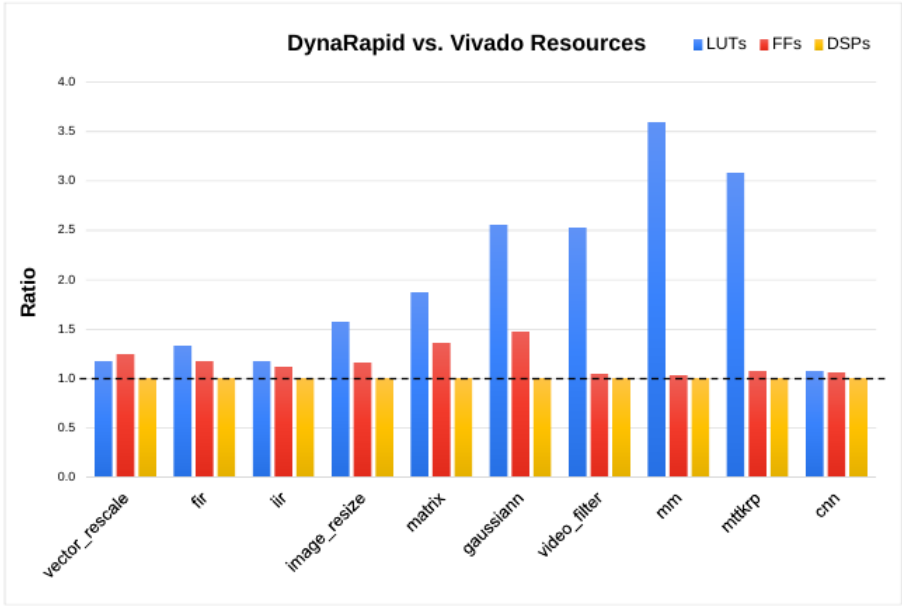
The benchmarks are typical HLS kernels

Benchmark	Components	Runtime			Fmax			LUTs			FFs			DSPs	
		DynaRapid	Vivado	speedup	DynaRapid	Vivado	ratio	DynaRapid	Vivado	ratio	DynaRapid	Vivado	ratio	DynaRapid	Vivado
vector_rescale	50	15	489	33	140	179	0.8	626	534	1.2	687	552	1.2	3	3
fir	65	15	482	32	235	345	0.7	845	637	1.3	953	813	1.2	3	3
iir	91	19	505	27	173	204	0.9	1306	1113	1.2	1838	1644	1.2	6	6
image_resize	113	21	497	24	137	154	0.9	2376	1339	1.5	1521	1318	1.3	0	0
matrix	167	33	524	16	147	143	1	3912	2090	1.8	3164	2341	1.5	3	3
gaussiann	178	21	495	24	138	149	0.9	3706	1455	2.5	2669	1810	1.0	3	3
video_filter	186	32	521	16	143	154	0.9	5985	2368	2.5	2969	2847	1.0	9	9
mttkrp	319	40	509	13	72	63	1.1	14487	4038	3.5	32222	3138	1.0	3	3
mm	201	26	474	18	95	85	1.1	6833	2226	3.0	1964	1832	1.1	6	6
cnn	790	50	524	10	152	160	0.9	10144	9449	1.1	12406	11731	1.1	18	18

Avg. runtime **20x** with **0.9x** Fmax



Avg. **1.8x LUTs; 1.2x FFs; 1x DSPs**



Conclusion

- **Open-source design flow** targeting runtime first
  - Fully placed and routed circuit in as little as **15** seconds
  - Speedup of **20x** with only **10%** frequency penalty

DynaRapid Applications

- Simplify the access to FPGAs to software programmers
- Enable fast-prototyping with hardware in the loop

DynaRapid fundamentally changes the way to interact with FPGAs, generating placed and routed circuits in seconds

Thank you!

andrea.guerrieri@ieee.org

COPYRIGHT AND DISCLAIMER

The information presented in this document is for informational purposes only and may contain technical inaccuracies, omissions, and typographical errors. The information contained herein is subject to change and may be rendered inaccurate releases, for many reasons, including but not limited to product and roadmap changes, component and motherboard version changes, new model and/or product differences between differing manufacturers, software changes, BIOS flashes, firmware upgrades, or the like. Any computer system has risks of security vulnerabilities that cannot be completely prevented or mitigated. AMD assumes no obligation to update or otherwise correct or revise this information. However, AMD reserves the right to revise this information and to make changes from time to time to the content hereof without obligation of AMD to notify any person of such revisions or changes.

THIS INFORMATION IS PROVIDED 'AS IS." AMD MAKES NO REPRESENTATIONS OR WARRANTIES WITH RESPECT TO THE CONTENTS HEREOF AND ASSUMES NO RESPONSIBILITY FOR ANY INACCURACIES, ERRORS, OR OMISSIONS THAT MAY APPEAR IN THIS INFORMATION. AMD SPECIFICALLY DISCLAIMS ANY IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY, OR FITNESS FOR ANY PARTICULAR PURPOSE. IN NO EVENT WILL AMD BE LIABLE TO ANY PERSON FOR ANY RELIANCE, DIRECT, INDIRECT, SPECIAL, OR OTHER CONSEQUENTIAL DAMAGES ARISING FROM THE USE OF ANY INFORMATION CONTAINED HEREIN, EVEN IF AMD IS EXPRESSLY ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.