THE UNIVERSITY *of* EDINBURGH

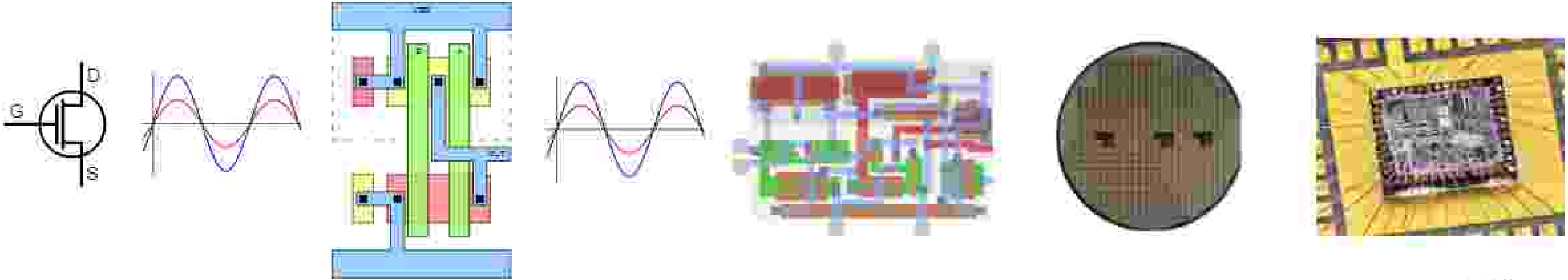
Acceleration of Integrated Circuit Simulation using SYCL and oneAPI

Danial Chitnis and Finlay Marno

Nov 2021

Simulation of Integrated Circuits

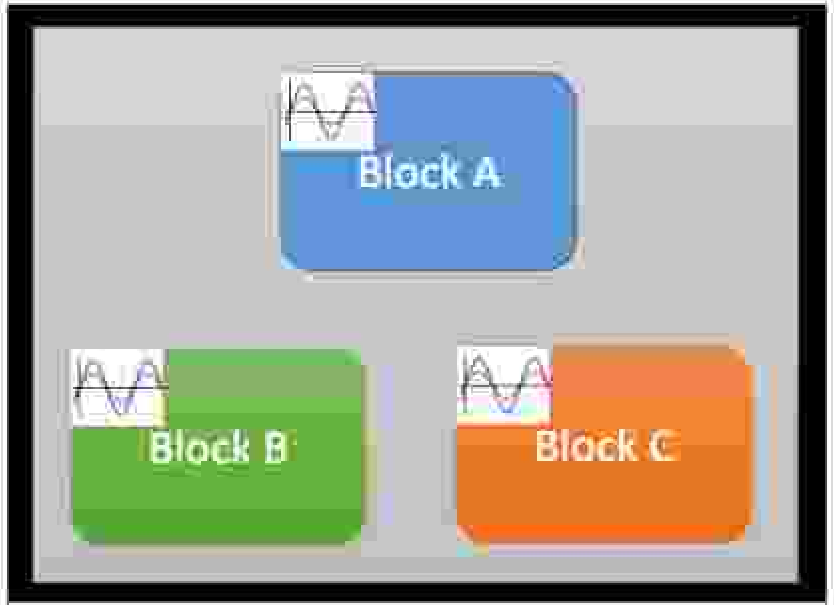
- **Integrated circuit design**
  connecting transistors – billions of them!
- **Simulation before fabrication**
  Chip fabrication is expensive and lengthy
- **The chip design process**

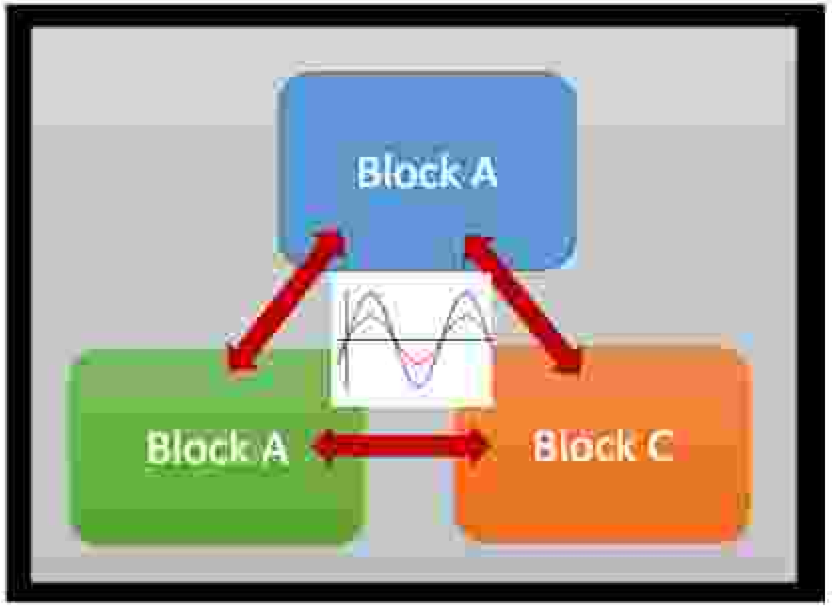


Simulation of Integrated Circuits

- The need for full-chip simulation
  - block level simulation
  - Higher abstraction levels – (e.g. processors)
  - The interaction between each block is often ignored

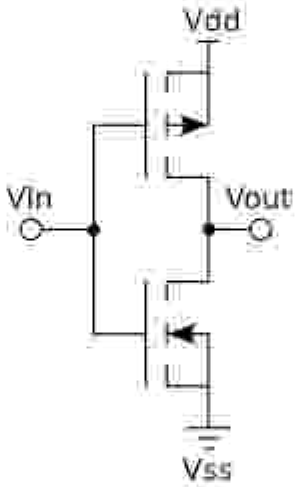
Run time examples: Phase Lock Loop circuit: ~1 week





Simulation of Integrated Circuits

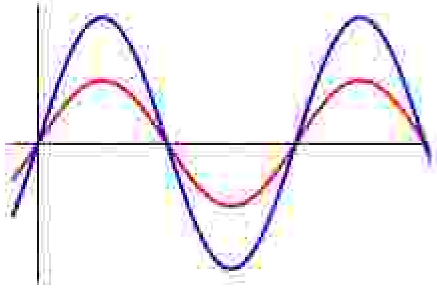
- Circuit Simulator is a complex programme:
  - Schematic entry
  - Netlisting
  - Parser
  - Solver
  - Result preparation
- We focus on the compute intensive “solver” in this work



```
Basic RLC circuit
.include modelcard.CMOS90
r vdd 2 100.0
l vdd 2 1
c vdd 2 0.01
m1 2 1 0 0 N90 W=100.0u L=0.09u
vdd vdd 0 1.8
```

$$\begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{bmatrix}$$

$$\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} + \frac{\partial^2 u}{\partial z^2} = 0.$$



SYCL and oneAPI

- Faster and more efficient circuit simulators
- Express parallelism in a mature and stable language
- SYCL is a good choice since it is based on standard C++
- Intel's oneAPI implementation enables us to run our code without modification on CPU and GPU

Hand over to my colleague **Finlay Marno**…

Circuit Simulation

Solving a circuit:

- Find the current and voltage over the components
- Non-linear equations – approximate to linear
- Differential equations – numeric analysis

Comes down to solving many linear systems of equations

Circuit Simulation

Linear systems of equations:

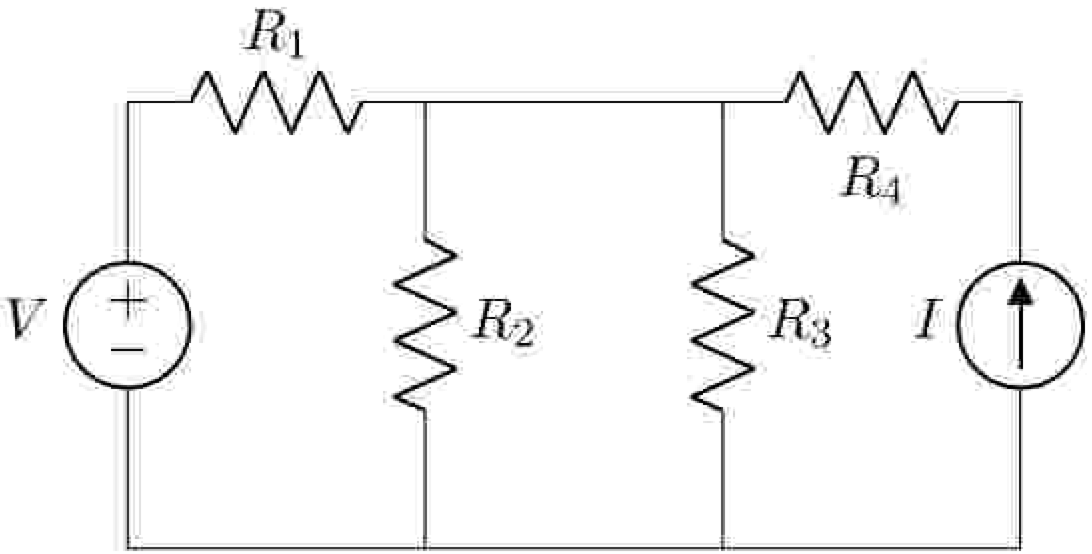
$$\begin{gathered} 6x + 2y + 8z = 26 \\ 3x + 5y + 2z = 8 \\ 0x + 8y + 2z = -7 \end{gathered}$$

$$\begin{bmatrix} 6 & 2 & 8 \\ 3 & 5 & 2 \\ 0 & 8 & 2 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} 26 \\ 8 \\ -7 \end{bmatrix}$$

Circuit Simulation

Create the system of equations using:

- Kirchhoff's Current Law
- Kirchhoff's Voltage Law
- Equation of the components (Ohm’s law $V = IR$ for linear resistors etc)

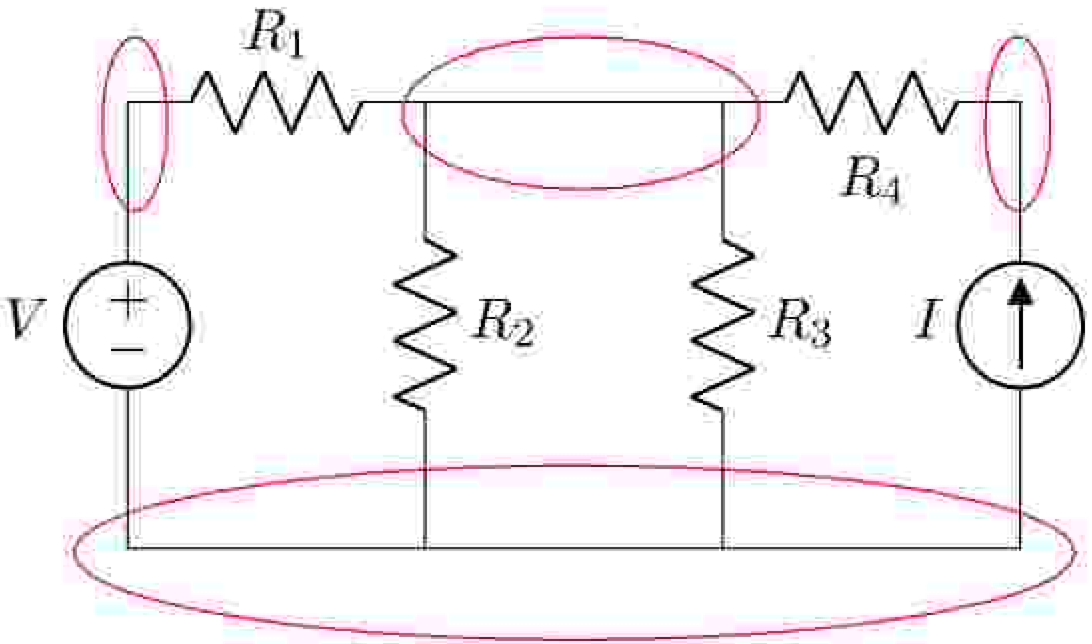


$$\begin{bmatrix}
1 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
-1 & 1 & 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & -1 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & -1 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & -1 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & -1 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 \\
R_1 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & R_2 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & R_3 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & R_4 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0
\end{bmatrix}
\begin{bmatrix} i_1 \\ i_2 \\ i_3 \\ i_4 \\ i_5 \\ i_6 \\ u_1 \\ u_2 \\ u_3 \\ u_4 \\ u_5 \\ u_6 \\ v_1 \\ v_2 \\ v_3 \end{bmatrix}
=
\begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ V \\ I \end{bmatrix}$$

Circuit Simulation

$$\boldsymbol{Number\ of\ rows = 2 \times number\ of\ components + number\ of\ nodes\ - 1}$$

Nodes are where components meet (real wires are more like resistors)



$$\begin{bmatrix}
1 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
-1 & 1 & 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & -1 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & -1 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & -1 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & -1 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 \\
R_1 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & R_2 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & R_3 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & R_4 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0
\end{bmatrix}
\begin{bmatrix} i_1 \\ i_2 \\ i_3 \\ i_4 \\ i_5 \\ i_6 \\ u_1 \\ u_2 \\ u_3 \\ u_4 \\ u_5 \\ u_6 \\ v_1 \\ v_2 \\ v_3 \end{bmatrix}
=
\begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ V \\ I \end{bmatrix}$$

LU Decomposition

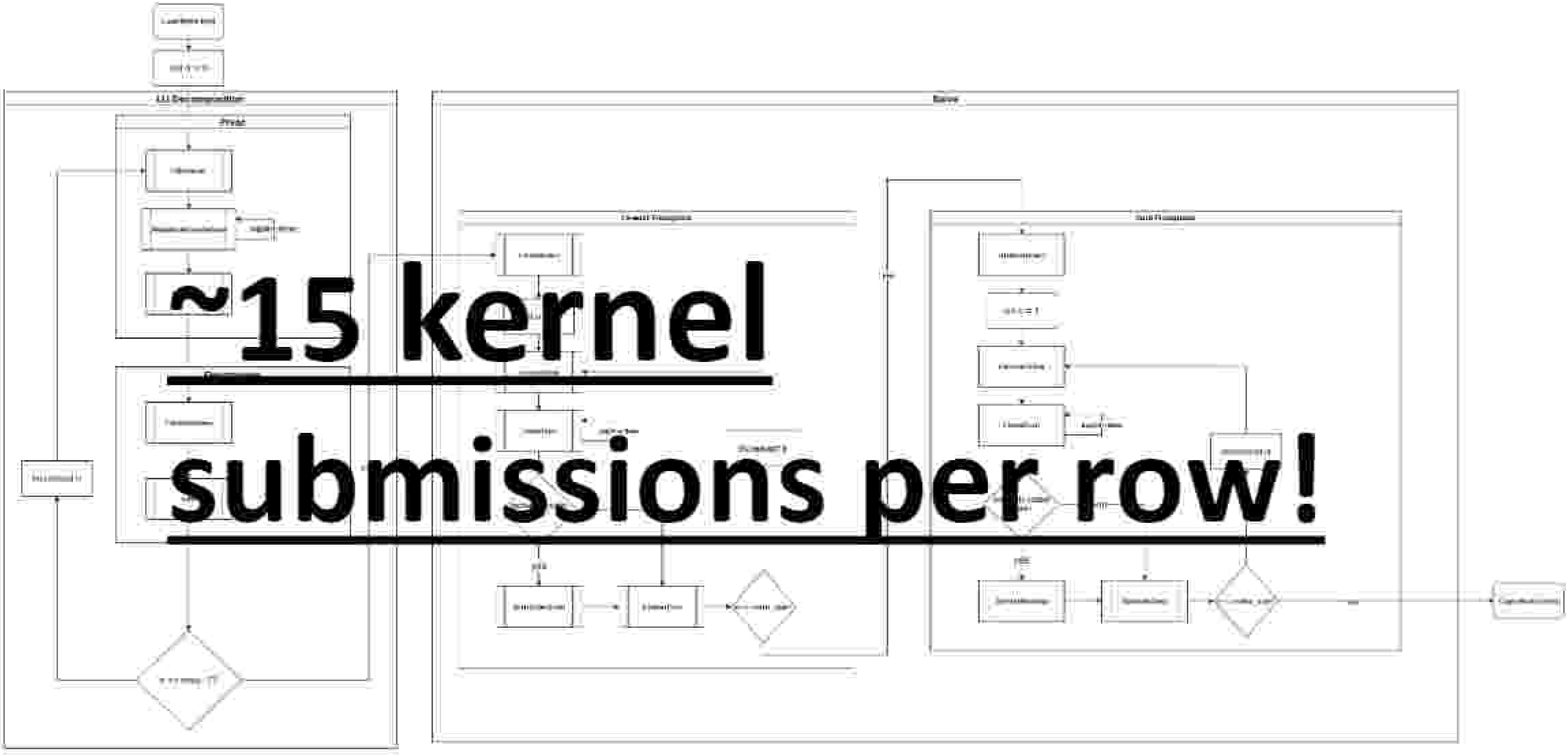
$$Ax = b$$

$$LUx = b$$

$$Ly = b\ Ux = y$$

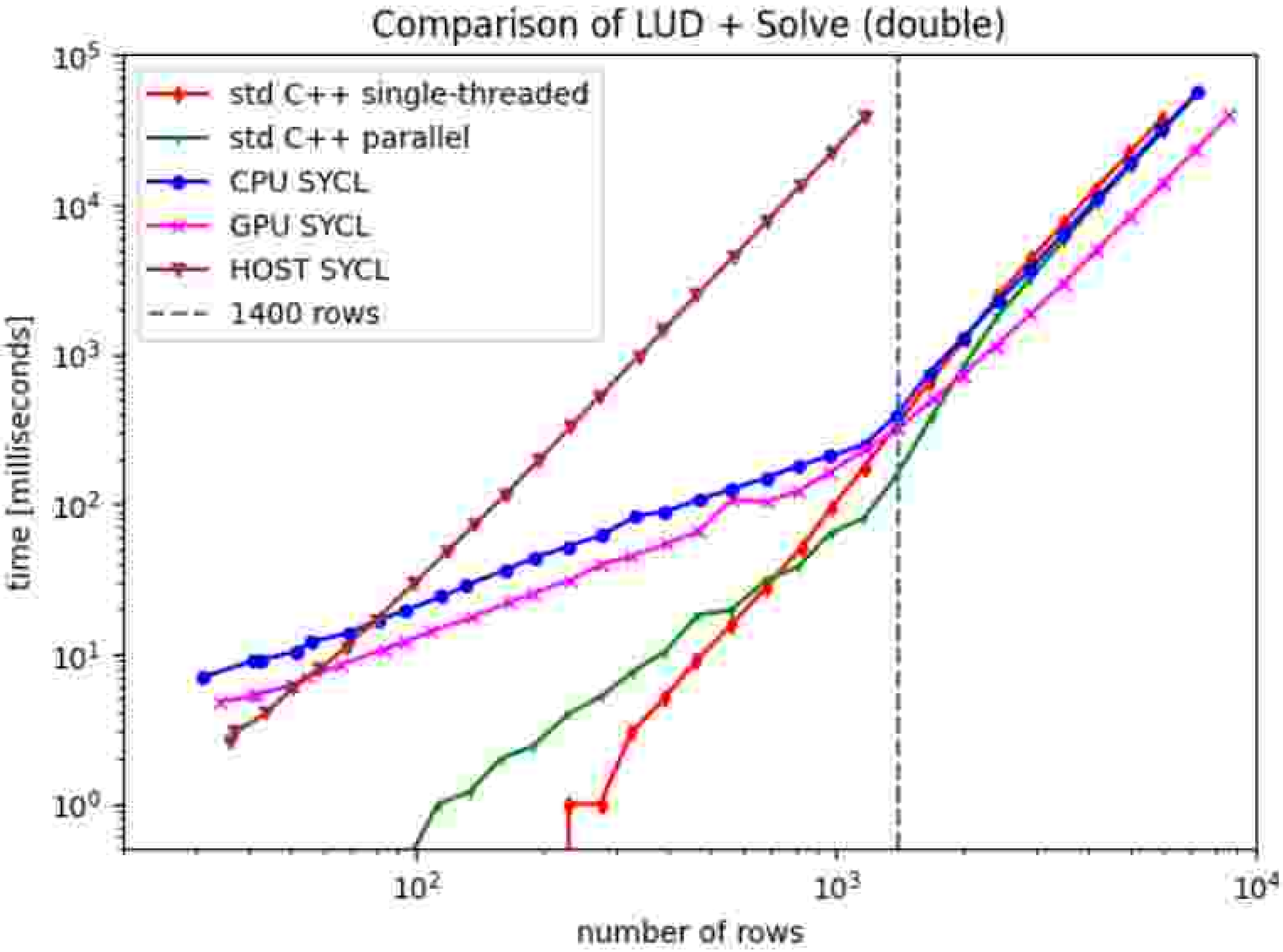
$$\begin{bmatrix} a_{11} & a_{21} & a_{31} \\ a_{12} & a_{22} & a_{32} \\ a_{13} & a_{23} & a_{33} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ l_{12} & 1 & 0 \\ l_{13} & l_{23} & 1 \end{bmatrix} \begin{bmatrix} u_{11} & u_{21} & u_{31} \\ 0 & u_{22} & u_{23} \\ 0 & 0 & u_{33} \end{bmatrix}$$

LU Decomposition

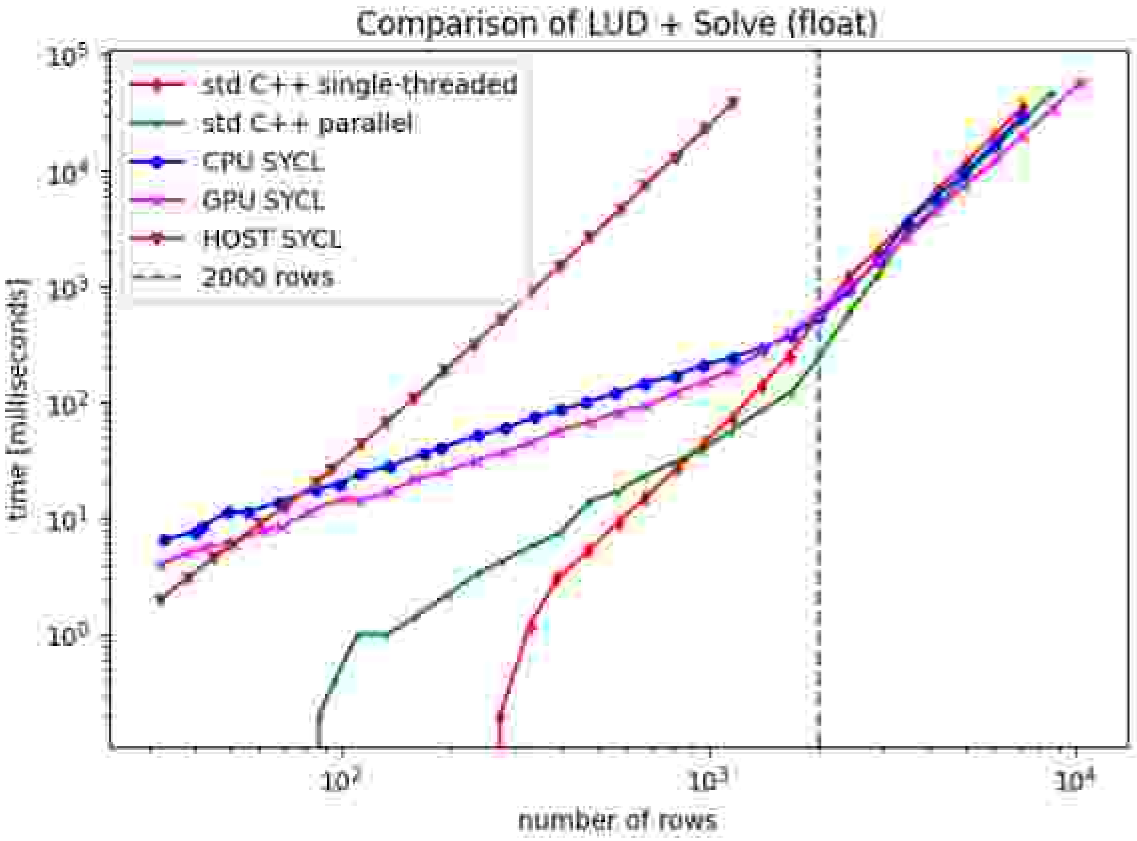


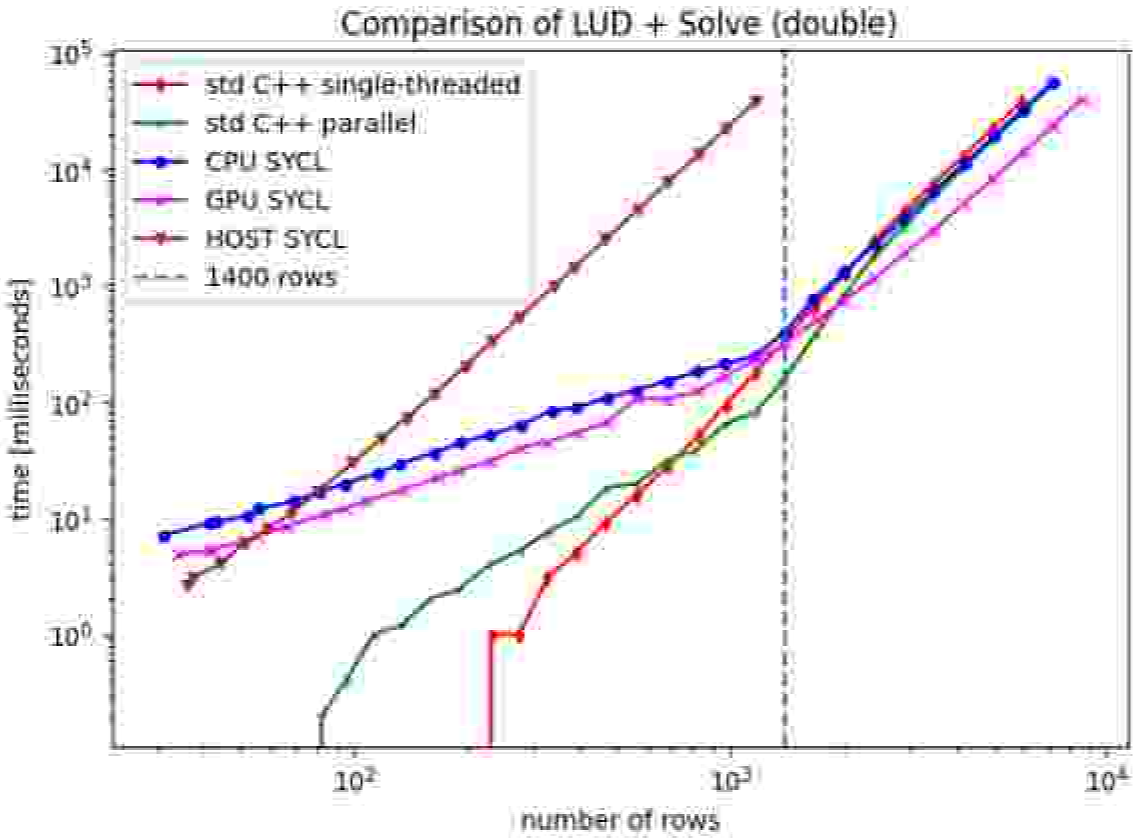
~15 kernel submissions per row!

Performance Comparison



Performance Comparison





Performance Comparison

It runs it faster!

Time to celebrate?

maybe not...

Energy Comparison

Implementation	Float Performance Ratio	Double Performance Ratio	Power Usage	Float Energy Ratio	Double Energy Ratio
Sequential	1	1	30W	1	1
Parallel STL	1.21	1.13	55W	1.52	1.63
SYCL CPU	1.05	1.09	55W	1.74	1.68
SYCL GPU	1.58	2.51	110W	2.32	1.46

Faster

More Energy

$$energy\ ratio = \frac{P_{other}}{perf\ ratio \times P_{cpu}}$$

*Power draw measured using Open Hardware Monitor - https://openhardwaremonitor.org/

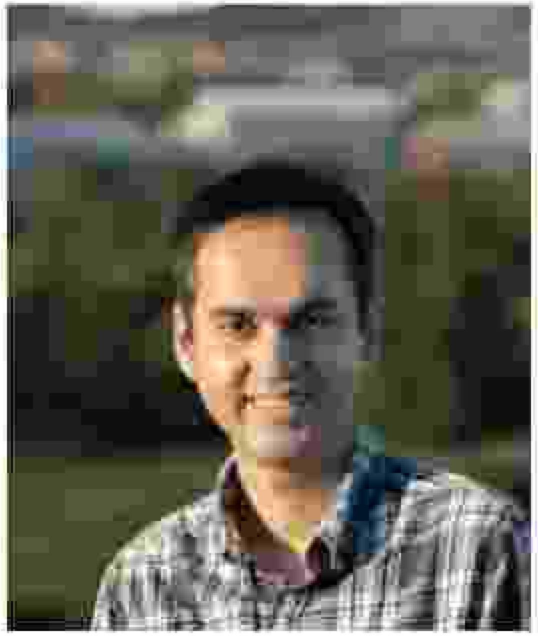
Conclusions

- Parallelising LU Decomposition is non-trivial
- SYCL made getting started easy
- Same code can run on multiple devices without modification
  - Intel(R) Core(TM) i7-9700
  - NVIDIA RTX A4000
  - Intel(R) Xeon(R) E-2176G CPU
  - Intel(R) Iris(R) Xe MAX Graphics
- Top performance is still not easy
- Energy is an important factor

Future work

- Comparison with other GPU frameworks
- Wait for SYCL to get better?
- Comparison with FPGA
- More complex circuits
- Artificial Intelligence
- Different solving algorithms

Thank You!



Dr Danial Chitnis

d.chitnis@ed.ac.uk

THE UNIVERSITY
of EDINBURGH

Finlay Marno*

f.s.marno@sms.ed.ac.uk
finlay.marno@gmail.com

**Now with Codeplay Software*