# Temperature-aware Hybrid Power Models in Embedded GPUs

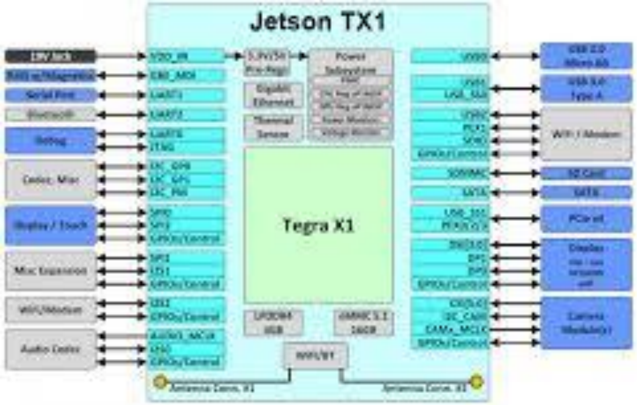
*Jose Nunez-Yanez, Kris Nikov,*
*Kerstin Eder, Mohammad Hosseinabady,*
*University of Bristol, UK*

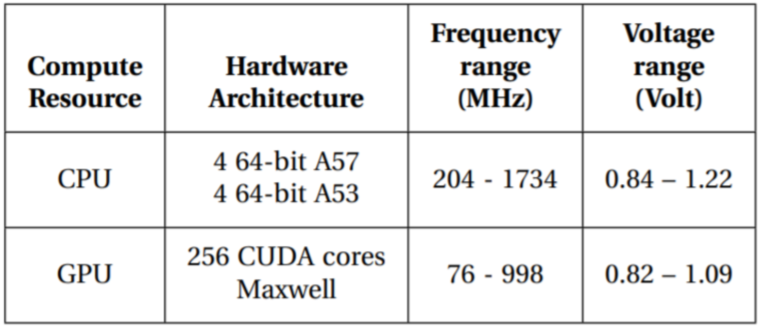
# Talk structure

1. Introduction to Tegra TX1 SoC and power modelling methodology.
2. Power modelling with per-frequency and unified models.
3. Understanding the impact of device temperature on power predictions.
4. Conclusions and future work.

# Tegra TX1 MPSoC

- Heterogenous device with CPU and GPU compute resources.
- Previously we have investigated power modelling on the big.LITTLE CPU and in this work we focus on the GPU.
- Methodology should also be applicable to TX2 and Xavier.





| Compute Resource | Hardware Architecture | Frequency range (MHz) | Voltage range (Volt) |
|---|---|---|---|
| CPU | 4 64-bit A57<br>4 64-bit A53 | 204 - 1734 | 0.84 – 1.22 |
| GPU | 256 CUDA cores Maxwell | 76 - 998 | 0.82 – 1.09 |

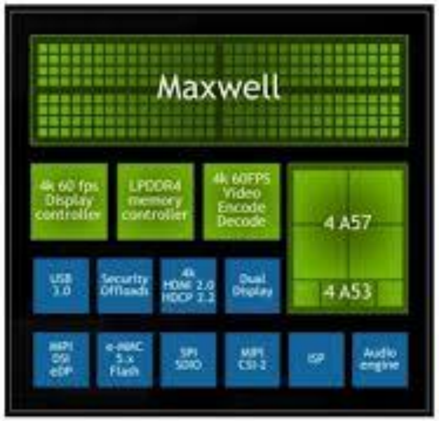
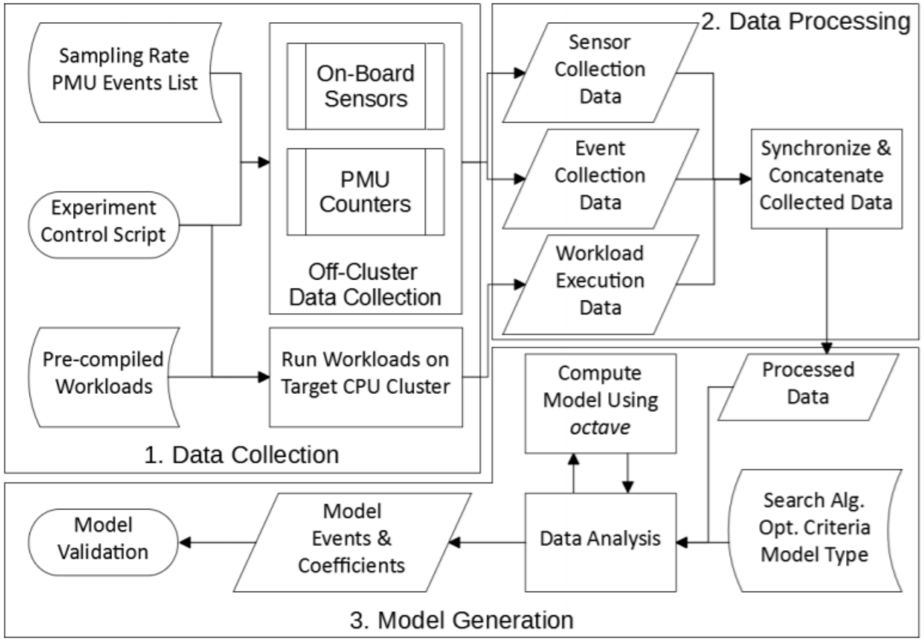


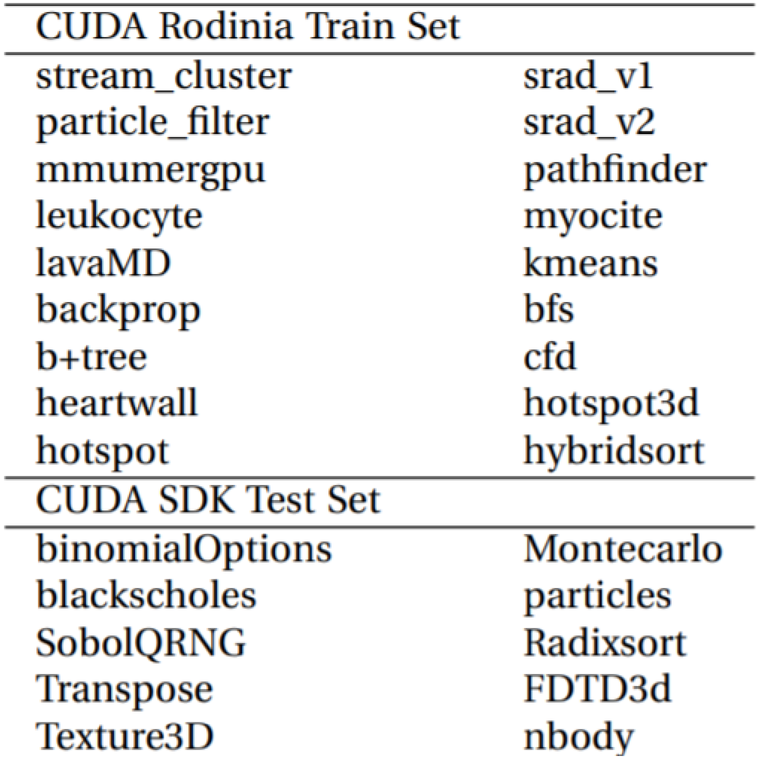
Figure 5 NVIDIA Tegra X1 Mobile Processor

Tegra TX1 hardware details

# Power modelling methodology overview

- Data collection is board/device dependent.
- Synchronization needed to synchronize events with power sensors when multiple runs are used to collect sensor data.
- Independent CUDA benchmarks for linear regression train and test phases.

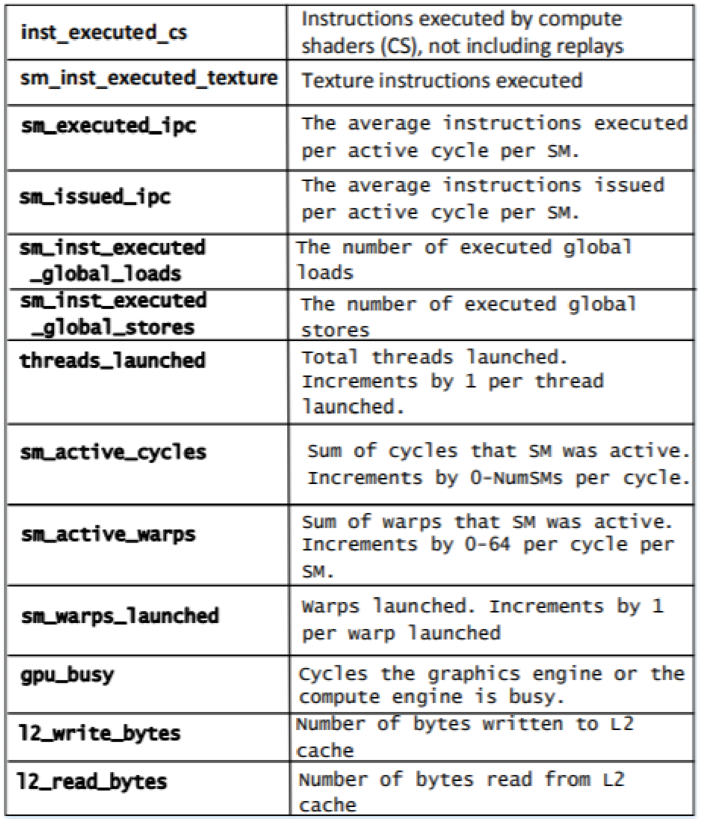




| CUDA Rodinia Train Set | |
|---|---|
| stream_cluster | srad_v1 |
| particle_filter | srad_v2 |
| mmumergpu | pathfinder |
| leukocyte | myocite |
| lavaMD | kmeans |
| backprop | bfs |
| b+tree | cfd |
| heartwall | hotspot3d |
| hotspot | hybridsort |
| **CUDA SDK Test Set** | |
| binomialOptions | Montecarlo |
| blackscholes | particles |
| SobolQRNG | Radixsort |
| Transpose | FDTD3d |
| Texture3D | nbody |

# Performance counter selection and analysis

- Pre-selection of 13 performance counters based on user experience.
- Example commands to find best model with 4 counters and calculate model with 4 particular counters across all the available frequency/voltage points.



| | |
|---|---|
| inst_executed_cs | Instructions executed by compute shaders (CS), not including replays |
| sm_inst_executed_texture | Texture instructions executed |
| sm_executed_ipc | The average instructions executed per active cycle per SM. |
| sm_issued_ipc | The average instructions issued per active cycle per SM. |
| sm_inst_executed _global_loads | The number of executed global loads |
| sm_inst_executed _global_stores | The number of executed global stores |
| threads_launched | Total threads launched. Increments by 1 per thread launched. |
| sm_active_cycles | Sum of cycles that SM was active. Increments by 0-NumSMs per cycle. |
| sm_active_warps | Sum of warps that SM was active. Increments by 0-64 per cycle per SM. |
| sm_warps_launched | Warps launched. Increments by 1 per warp launched |
| gpu_busy | Cycles the graphics engine or the compute engine is busy. |
| l2_write_bytes | Number of bytes written to L2 cache |
| l2_read_bytes | Number of bytes read from L2 cache |

```
octave_makemodel.sh -r measurement.txt -
    b benchmark.txt -f
    76,153,230,307,384,460,537,614,691,
    768,844,921,998 -p 7 -m 1 -l
    8,9,10,11,12,13,14,15,16,17,18,19,20
    -n 4 -c 1 -o 2
```

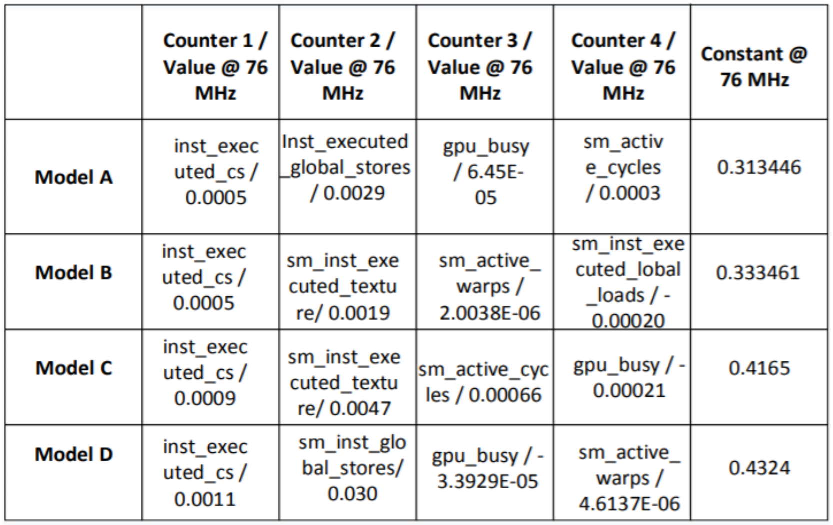
```
./octave_makemodel.sh -r
    power_measurement.txt -b benchmark.
    txt -f
    76,153,230,307,384,460,537,614,691,
    768,844,921,998 -p 7 -e 8,11,13,19 -o
     2
```

# Per-frequency models

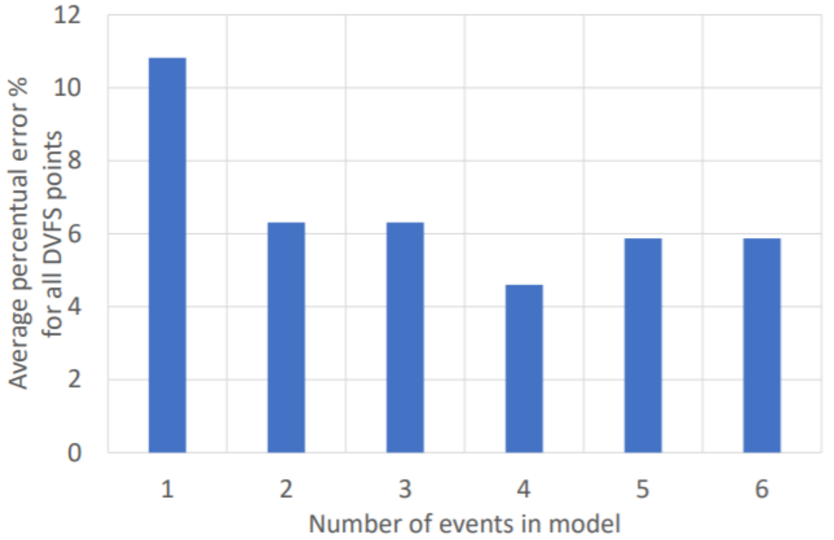
- General form of the considered model with events normalized by clock cycles.

$$P_{GPUfreq_1} = \alpha_0 + \alpha_1 \times events_1/cycles + \ldots + \alpha_n \times events_n/cycles$$

- Multiple linear regression calculates the α coefficients with one constant used to capture idle power.
- The per-frequency model has a different set of coefficients for each voltage/frequency pair. Table shows example at frequency 76 MHz and voltage 0.82V



| | Counter 1 / Value @ 76 MHz | Counter 2 / Value @ 76 MHz | Counter 3 / Value @ 76 MHz | Counter 4 / Value @ 76 MHz | Constant @ 76 MHz |
|---|---|---|---|---|---|
| **Model A** | inst_executed_cs / 0.0005 | Inst_executed_global_stores / 0.0029 | gpu_busy / 6.45E-05 | sm_active_cycles / 0.0003 | 0.313446 |
| **Model B** | inst_executed_cs / 0.0005 | sm_inst_executed_texture/ 0.0019 | sm_active_warps / 2.0038E-06 | sm_inst_executed_lobal_loads / -0.00020 | 0.333461 |
| **Model C** | inst_executed_cs / 0.0009 | sm_inst_executed_texture/ 0.0047 | sm_active_cycles / 0.00066 | gpu_busy / -0.00021 | 0.4165 |
| **Model D** | inst_executed_cs / 0.0011 | sm_inst_global_stores/ 0.030 | gpu_busy / -3.3929E-05 | sm_active_warps / 4.6137E-06 | 0.4324 |



Event number vs Error

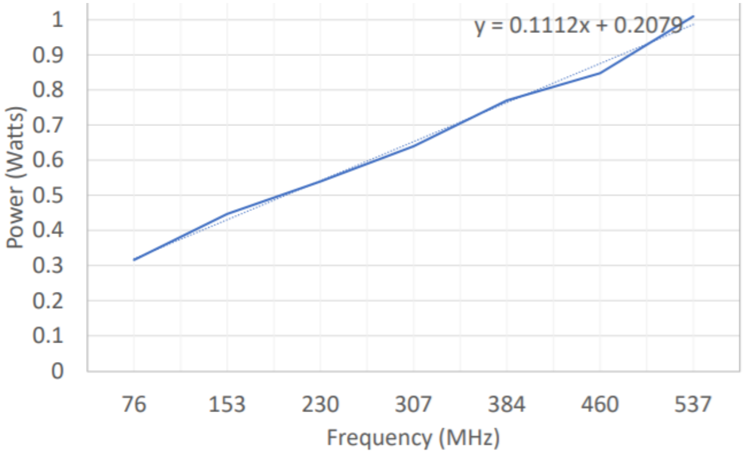
# Unified models

- Can we use a single set of coefficients and scale the power to other frequency and voltage points ? Less data to handle.
  - This takes into account that for each voltage level multiple frequency points are possible according to the DVFS table.
  - Isolating static power is also useful to predict the temperature impact on static power (no effect dynamic power).

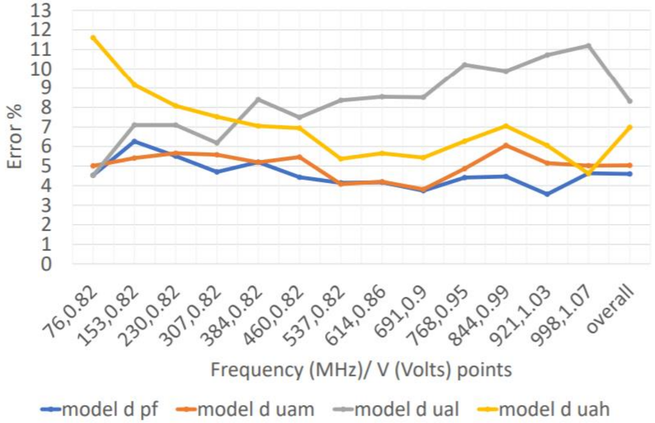
$$P_{dynamic\_clock} = alpha \times C \times V^2 \times f$$

$$P_{idle} = P_{dynamic\_clock} + P_{static}$$

$$P_{GPUfreq_x} = (P_{GPUfreq_1} - P_{GPUsta_x}) \times \frac{freq_1}{freq_x} \times (\frac{volt_1}{volt_x})^2 + P_{GPUsta_x} \times (\frac{volt_1}{volt_x})^2$$



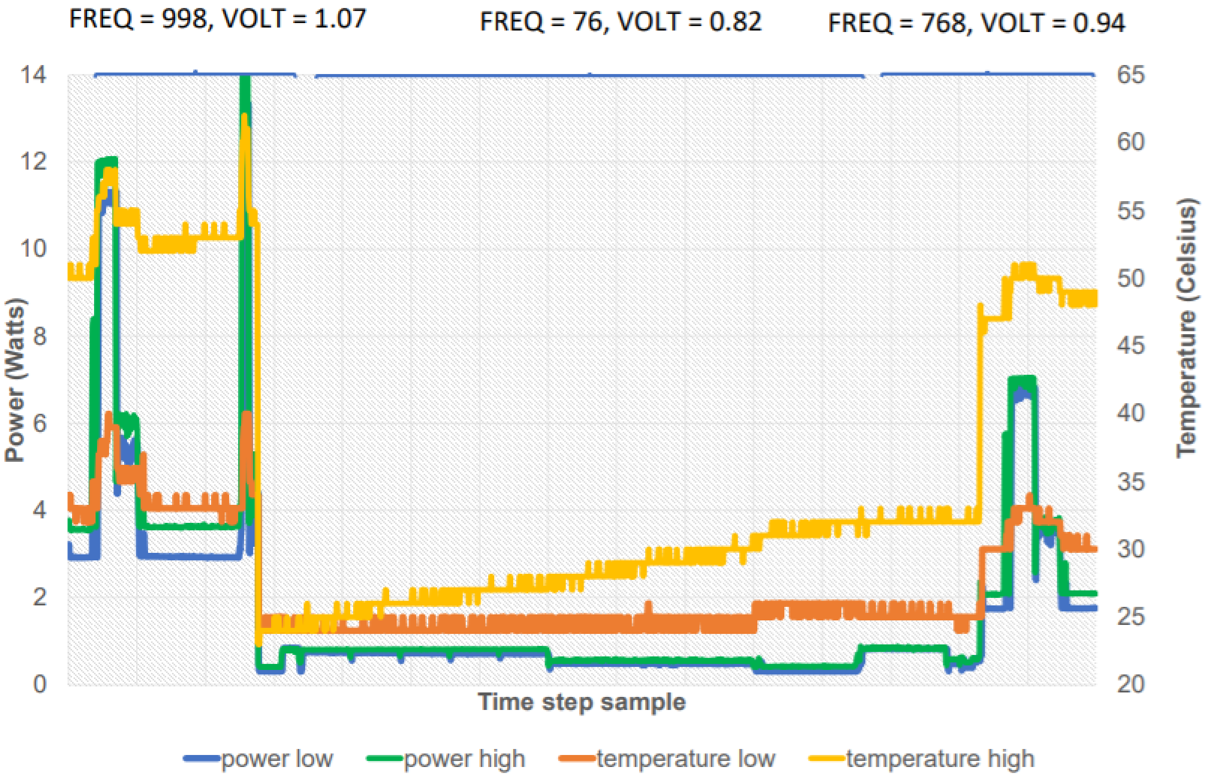
Tegra TX1 idle power



Model error comparison

# Temperature and power analysis

- Understanding the effects of temperature in power.
  - We observe up to 20% power variation between 55C (fan off) and 35C (fan on) temperature operation.
  - Specially relevant for fanless deployments of the device.

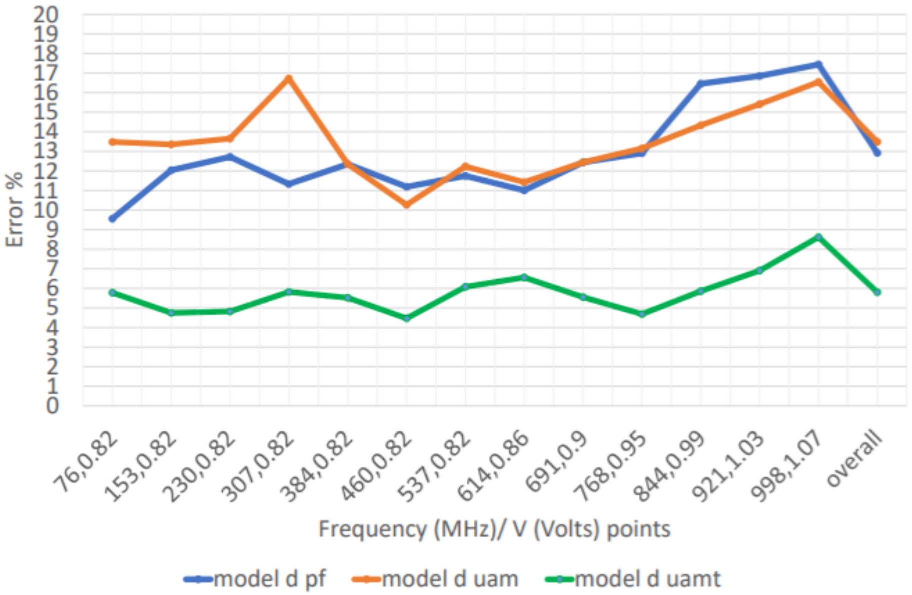


Tegra TX1 temperature effect on power

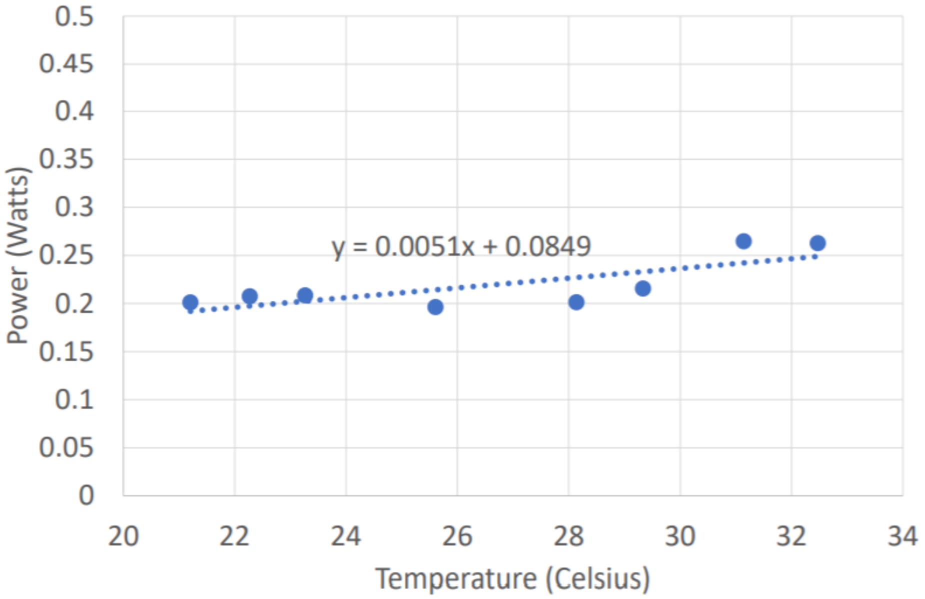
# Temperature-aware power model

- Extending unified power model to account for temperature scaling.
- Linear relation between static power and temperature scaled to different voltage levels.
- Error at high temperature maintained around 5%.

$$P_{GPUfreq\_x} = (P_{GPUfreq\_ref} - (Tref \times 0.0051 + 0.0849)W) \times \frac{freq\_x}{380MHz} \times (\frac{volt\_x}{0.82V})^2 + (T \times 0.0051 + 0.0849)W \times (\frac{volt\_x}{0.82V})^2$$



Temperature-aware power model



Temperature impact on static power

# Conclusions

- Tegra SoC TX1 GPU instrumented with power measurement and performance counter features and a power model developed based on multiple-linear regression.
- Extended to multiple voltage and frequency points with per-frequency and unified models accuracy of around 5% executing varied CUDA benchmarks.
- The hybrid unified model uses local events (i.e. performance counters) and global states (i.e. voltage, frequency and temperature) to obtain a general solution.
- This general solution eliminates temperature induced prediction errors of up to 20% and accounts for the multiple frequency points possible for a single voltage level.

# Acknowledgement

- Thanks to EPSRC and for the support with the ENEAC projects, Royal Society with MINET project and the H2020 TeamPlay project.
- Questions ?