CSE-1420

Date 10/10/2024

Embedded Systems

→ TPU, NPU

→ CPU and Memory

→ MAC units

$$c = a + b$$
$$d = c + f$$

→ Ultimate bottleneck is memory.

→ Harvard Arch. → seperate data and instruction.

→ Dataflow Arch.

→ 10-100 ms

→ Main Memory is based on DRAM.

L1 → 1 cycle
L2 → few "
L3 → few 10 cycles

$$\boxed{\begin{aligned} c &= a+b \\ d &= c+f \end{aligned}}$$

temporal locality

→ Memory locality

Spatial

$$\begin{bmatrix} \overrightarrow{x_{00} \; x_{01} \; x_{02} \cdots} \end{bmatrix} \begin{bmatrix} y_{00} & \cdot \\ y_{01} & | \\ y_{02} & | \\ \vdots \downarrow & | \end{bmatrix}$$

→ pre-fetching

L1 is subject of L2
L2 is " " L3

→ DRAM
→ DIMM Module

channel, ranks, banks

rank – 64 B data
→ bus width

→ 64 ms refresh rate of DRAM

→ Sense Amplifier → first sense then amplify
→ peripheral circuitry
→ A buffer

→ row buffer locality

→ NVMain
→ DRAMSIM
→ Ramulator

→ closed row policy
reads are destructive (then restore dat)

→ open row policy
to avoid precharging

→ order of commands → typical flow for every command.
ACT
RD
PRE

24/10/2024

Take 1K x 1K matrix.

L2 -> 1MB

INT 4B

4B x 1K

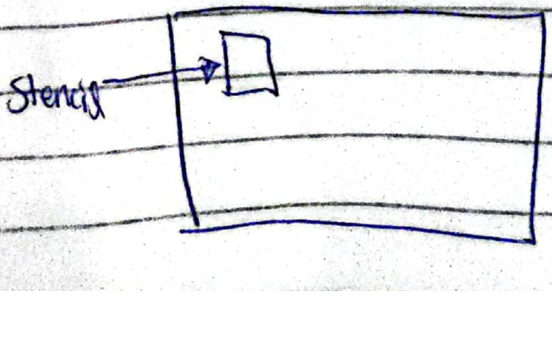
Then note the execution time.

By tiling we can exploit the temporal locality

```
For(i=1-100){
    for(j=1-100){
```

```
for(i=i-50)
    for(j=
```

-0 and -3



0 - 1 Black & White
0 - 255 Grayscale.

mucol

① NVsim
② CACTI

HBM → High Bandwidth Channel

64B →	4B
1 w/s	16 w/s

→ Decompression is on critical path

→ Latency & Throughput is same for non-pipelining system.

→ OOO processor

→ In order, Keep pipeline in order.

→ Arm A15

SISD, SIMD

$$\begin{bmatrix} A_{00} & A_{10} \\ & \end{bmatrix} \begin{bmatrix} b_{00} \\ b_{10} \end{bmatrix}$$

GPUs are SIMD

→ GPT 4 training resources

DSL → Domain Specific language.

→ Scratchpad memory.
↳ equivalent of cache but is Software managed.

<u>HMC</u> → multiple stack layers.

Assignment #1
DRAM timing parameters.

TACT
upto 20 different commands.

Date 31/10/2024 ES

- → GP

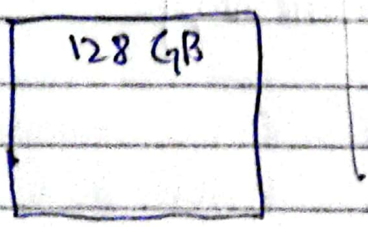
- → Compute bound
- → Memory 〃 (bounded by load store)

Ops/byte ↑↓

- → SPEC 2017 → benchmark for Compute bound/ Memory 〃

- → Near Memory Computation

- → Two Categories ① CIM ② CNM



- → Cost Model.
- → must read " The landscape "
- → french company upmem

- → DRAM near computation (1970s)
- → MRAM (Main RAM)
- → WRAM (Working 〃)

5000 DPUS

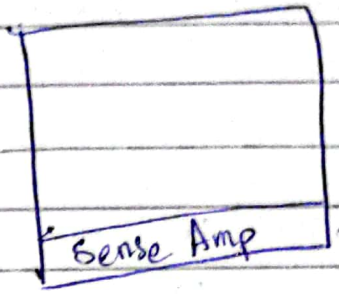
- → Software Emulated
  - ↳ Software toolchain converts it int.
- → Synchronous / Asynchronous.
- → GCC LLVM has own IR.
- → MLIR → Multilevel IR
- → Cinnamon flow.
- → BLAS → baseline for optimization.
- → Command for broadcast., scatter and gather
- → Alveo also uses HBM memory.
- → AiM → 2 neighbor banks can communicate.
- → GDDR & compromized b/w HBM and HMC)
- → In Memory Computing Next class.
  DRAM, SRAM.

Date 7/11/2024

Crossbar

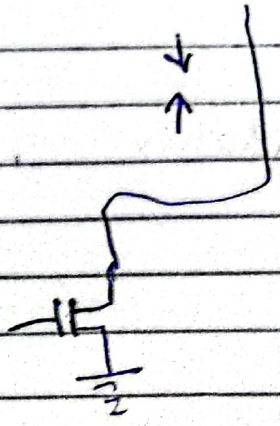
→ O(n^3) — O(1)

Word size 512 - 4KB



CAMs → Good for searching.

→ SRAM are fastest



800K	100K
LRS	HRS
1	0

800K	400K	200K	100K
11	10	01	00

→ Low Resistance State

High " "

→ Racetrack Memory → Similar to HDD

→ writing is extremely slow.

→ Endurance how many times a cell can be read/written.

→ Mem resistor

$$\begin{bmatrix}1 & 0 & 1\\ 0 & 1 & 0\\ 1 & 0 & 1\\ 0 & 1 & 0\end{bmatrix} g \qquad \begin{bmatrix}1\\ 0\\ 1\\ 0\end{bmatrix} V$$

writing entire Matrix

→ kcl

→ ISAA C

Shift and hold.

→ PUMA Sim

→ Sim-P