Conan Barbála 22MOTE

① Rent's rule

$N_p = K_p \cdot N_g^{\beta}$

where:
- N_p = number of I/Os of a certain logic block
- N_g = number of gates in a block
- β - Rent constant
- K_p the coefficient

We can calculate the number of signal pins of a certain logic block based on the number of gates in the block

② Signal delays on PCBs and on ICs + explain

To switch from IC to PCB means a great loss of time, the capacitive ratio of PCA : ICA is 1000 : 1

IC: $C_u \approx 1,8\ pF/cm$ $\quad l \approx 10\mu m$

PCB: $C_u \approx 2\ pF/cm$ $\quad l \approx 1\ cm$

$(C_L = l \cdot C_u)$

So we can use this formula, because the IC can be modelled by the concentrated capacitances

The higher the integrity, the higher the role of the interconnecting wires on signal propagation

③ Multiplying two matrices by using systolic processor arrays

Structure + data stream

if we have to matrices: A and B $(w \times w)$

bandwidth $\rightarrow w_1$

$$\left\{\begin{bmatrix} & & \\ & & \\ & & \end{bmatrix}\right. \quad \left\{\begin{bmatrix} & & \\ & & \\ & & \end{bmatrix}\right. \quad \rightarrow \underline{\underline{C}} = \underline{\underline{A}} \cdot \underline{\underline{B}}$$

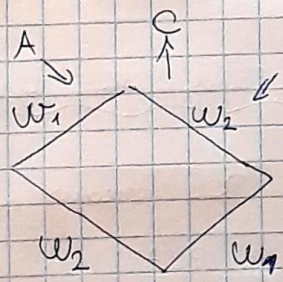
$\underline{\underline{A}}$ $\underline{\underline{B}}$

we can compute c_{ij} by the following formula:

$$c_{ij}^{(0)} = 0$$

$$c_{ij}^{(k+1)} = c_{ij}^{(k)} + a_{i,k} \cdot b_{k,j}$$

we will have $\underline{w_1 \cdot w_2}$ processors, they will be in a hexagonal grid



time: $T = 3w + \min(w_1, w_2)$

The processors are active in every third time step.

④ Analog computation model

$\dot{x}(t) = f(x(t), u(t))$ (next state)

$y(t) = g(x(t), u(t))$ (output)

$x(t) \in C^2$ (the electrical signals have continuous values)

The analog computation model can be used for parallel computation because $y(t)$ and $y(t-1)$ are independent of each other → this way we can compute the output parallely.

$x(u)$

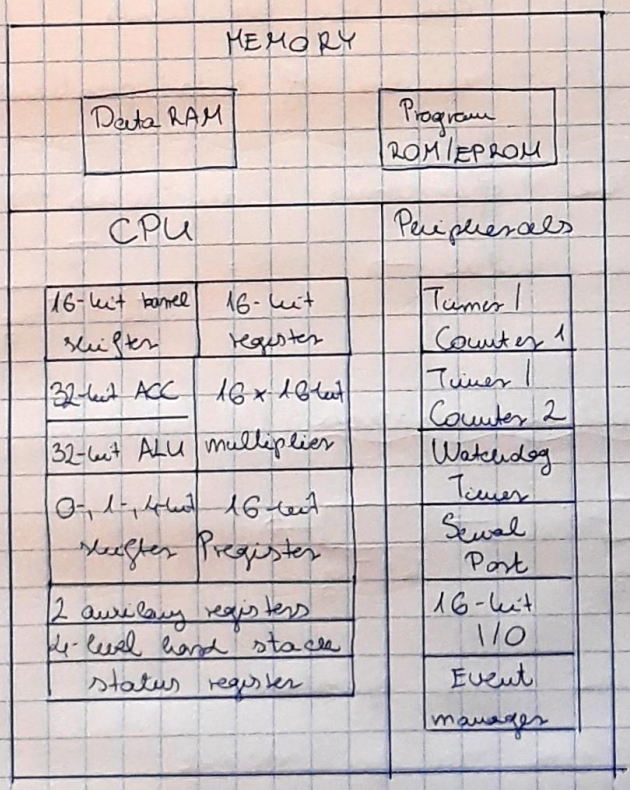
if we can also calculate $x(t+u)$,

(5) Parameters describing computational complexity

- k: the number of I/O pins
- A: the area of the I/O device (cm^2)
- E: energy (J)
- F: operating frequency (MHz)
- T: the time from the input to the output (ns)
- P: dissipation (W)

(6) Fixed point DSP structure

320 C14



Parts supporting parallel operation:

- 16×16 bit multiplier
- 16 bit barrel shifter
- 0-, 1-, 4- bit shifter

32-bit ALU: arithmetic + logical operations

receives input from the ACC, 16-bit P register

16×16 bit multiplier computes the 32-bit product

Registers: automatic incrementation, decrementation

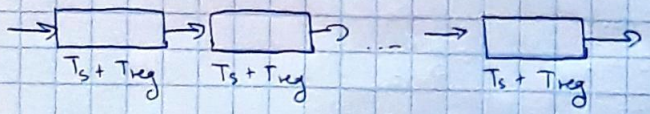
⑦ Time gain in pipeline operation

$$S = \frac{M \cdot T_p}{(M+N-1)(T_s + T_{reg})}$$

where:
- S: speedup
- T_p: time of the operation without the pipelining
- T_s: the runtime of the sub-operation
- T_{reg}: additional time due to data movements
- M: the number of times we run the operations
- N: number of sub operations

If we run the operations many times → M will have a large value. → $S = \frac{T_p}{T_s + T_{reg}}$

⑧ Bounds of pipeline operation



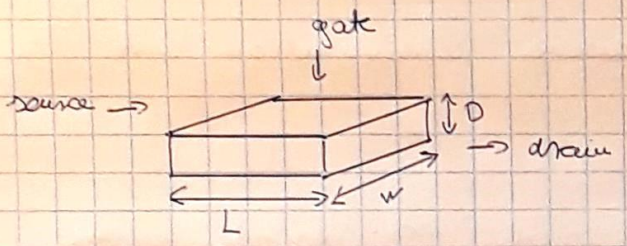
T_s time of one single operation

T_{reg} time of data transfer / storage

The slowest stage determines the maximum operation time.

maximum T_s: one result per clock cycle

(9) Effect of scaling down to gate capacity



we scale down L, W, D by 'k'

switching time: $\tau' = \frac{\tau}{k}$

dissipated power: $P'_{sw0} = \frac{P_{sw0}}{k^2}$

dissipated energy $E'_{sw0} = \frac{E_{sw0}}{k^3}$

By scaling down we improve the computational power.

(10) CNN mode with the basic equations.

$$\dot{x}_{ij}(t) = \sum A_{ij,kl}\, y_{kl}(t) + \sum B_{ij,kl}\, u_{kl} + z_{ij}$$

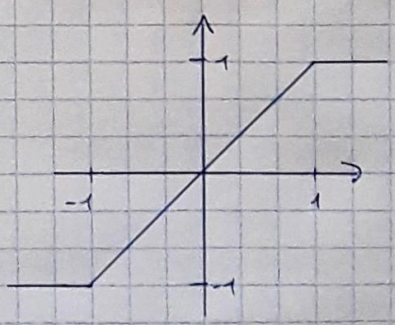
where: $A_{ij,kl}$, $B_{ij,kl}$ } template matrices

y_{kl} : output variable

u_{kl} : input variable

x_{ij} : state variable

z_{ij} : bias



discretized version of the equation

$$x_{ij}(k+1) = (1-a)\, x_{ij}(k) + a\left(\sum A_{ij,kl}\, y_{kl}(k) + \sum B_{ij,kl}\, u_{kl} + z_{ij}\right).$$

(11) Advantages of an open source ISA over commercial ISA

- the RISC-V supports extensive customization and specialization
- it uses only one additional user-visible register
- it has a minimal set of instructions
- less complex
- ability to redistribute