ZH

2021. október 16., szombat 18:53

1. Define the Rent rule (2)

Rent's rule

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

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

Give the number of signal pins of a certain logic block by knowing the number of gates inside the block.

2. Define the relationship of the signal delays on PCBs and on ICs, and give the explanation as well. (2)

The higher the integrity the higher the role of the interconnecting wires is on signal propagation! Two possible ways of modeling interconnections on ICs:

- Concentrated capacitances (concentrated parameter circuit model)
- Transmission lines (distributed parameter circuit model)

The transmission line effect is not significant if the wires are short or the changing of the signal is slow.

$C_L = \ell \cdot c_u$ where C_L: capacity of the wire
ℓ: length of the wire
c_u: capacity of a unit length wire

Typical data:

IC: $c_u \approx 1.8 \frac{pF}{cm}$ $\ell \approx 10\,\mu m$

PCB: $c_u \approx 2 \frac{pF}{cm}$ $\ell \approx 1\,cm$

PCB : IC typical capacitance ratio: 1000 : 1

$\Rightarrow$ to go from IC to PCB is a great loss of time. Just remember: $\tau = e \ln\left(\frac{C_L}{C_g}\right) \approx 30$

The loss

explanation as well. (2)
To multiply two matrices by using systolic processor arrays draw the structure and the data stream as well. How many processors have to be used? (2)
Which kind of operaton a processor doing? (0.5) Define the ara and the time requiremnts? (0.5)

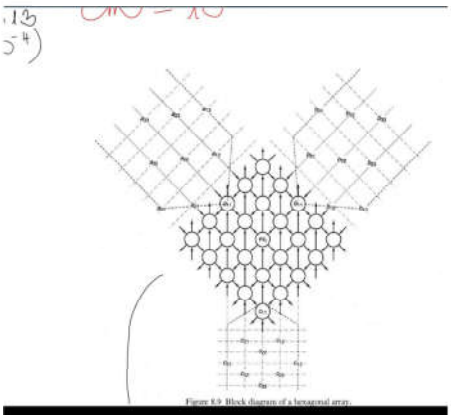
Multiplication of two matrices

Let us have:

- Two n x n band matrices A and B with bandwidth w1 and w2
- The task is to compute: C = A*B

Systolic array solution:

- Processors are in hexagonal grid structure
- Number of processors: w1*w2
- The recursive formula to compute: $c_{ij}^{(0)} = 0;\ c_{ij}^{(k+1)} = c_{ij}^{(k)} + a_{i,k} * b_{k,j};\ c_{ij} = c_{i,j}^{(n+1)}$
- $T = 3n + \min(w_1, w_2);\ A = w_1 w_2$
- The processors are active in each third-time step only



- The processors are active in each third-time step only

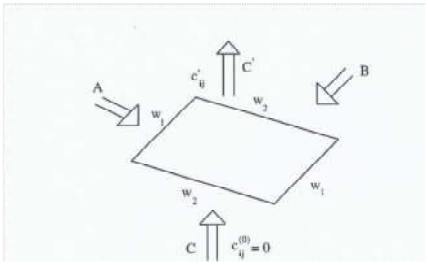
• Matrix – matrix multiplication

two $n \times n$ band matrices A and B
with bandwidths w_A and w_B

$$\begin{bmatrix} & & \\ & & \\ & & \end{bmatrix} \cdot \begin{bmatrix} & & \\ & & \\ & & \end{bmatrix} = \begin{bmatrix} & & \\ & & \\ & & \end{bmatrix}$$

HEXAGONAL arrangement

A processor is active in every THIRD timestep!



number of processors: $\# P = w_A \cdot w_B = A$ (area)
computation time: $3n + \min\{w_A, w_B\} = T$ (time)

Recursive formula to solve:

$$c_{ij}^{(0)} = 0$$
$$c_{ij}^{(k+1)} = c_{ij}^{(k)} + a_{ik} \cdot b_{kj}$$

4. Define the analog computation model . Explain why can it be used for parallel computation. (2)

Computation models

Computation is an operator: $f: u(t) \mapsto y(t)$

• Analog model

$$\dot{x}(t) = f(x(t), u(t))$$ ← "next state" (next is somewhat weird in continous time...)
$$y(t) = g(x(t), u(t))$$ ← output

$X(t) \in C^k$ (electrical signals with continuous values)

NOTE: Analog & Digital models can be parallelized: $y(t)$ does not depend on $y(t-1)$. Therefore the output can be calculated parallely. Also, if $x(k)$ is known, one can compute any $x(k+n)$ if $u(k+1), \ldots, u(k+n)$ are given. ⇒ Can run parallely.

Symbolic model can be described as a graph of an algorithm (flowchart). It can also contain parallelized parts.

Define the parameters describing the computational complexity. (2)

Typical parameters describing the computational complexity

A: area of the I/O device [cm²] [mm²]
E: energy needed to perform the function [J]
T: time from the start of input until the end of output [ns] [ps]
F: 1/(delay between input – next input) aka. operating frequency [MHz]
operations/second (reciprocal of F) [MIPS] [Flops] etc.
P: dissipation [W] [mW]
k: number of I/O pins
S: negentropy → The computation is a selection from the space of all possible outputs. $I \to O$, $|I| = X = |O|$ where the lengths: I: b bit long, X: p bit long, O: a bit long

In the calculation of the negentropy we select a specific 'a' long result from $2^b p$ states:

$$\Delta S_L = \log_2\left(\frac{\text{Size of the space of possible outputs [bit]}}{\text{Size of the result [bit]}}\right) = \log_2\left(\frac{2^b p}{a}\right)$$

If $p=a \Rightarrow \Delta S_L = b$
If $a=1 \Rightarrow \Delta S_L = \log_2(2^b p)$

Where *b* is the size of u(k) (the input), *p* is the size of x(k) (the state) and *a* is the size of y(k) (the result).

E_{SWO}: elementary switching energy, ~kT

- $\Delta E_{dis} \geq \Delta S_L * E_{SWO}$ energy is required to compute a computational operator.
- Moving the data to the computing units requires energy: In case of 1 bit it requires E_{SWO} energy, in case of *p* bits it requires pE_{SWO} energy. ($\Delta S_x = p$)
- The used wires need to be charged as well: $\left(\frac{l}{l_{SWR}}\right) * E_{SWO}$

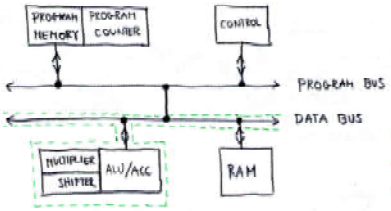
Energy efficiency:

$$\eta = \frac{(\Delta S_L + \Delta S_x + \Delta l / l_{SWR}) \cdot E_{SWO}}{E_{tx}}$$

6. Describe a fixed point DSP architecture, and define the function of the main blocks. Define the parts in the archetecture supporting the parallel operation. (4)

ext

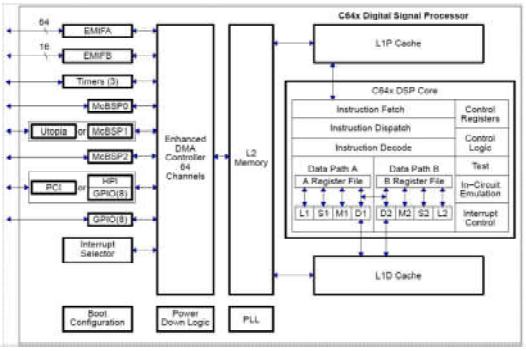
DSP architecture



- Advanced Harvard architecture
  - → Separate data/program buses
  - → Parallel { Instruction fetch, operand fetch, execute }
- Parallel multiplier
- Multiple shifters
- RAM
- Incrementing/decrementing registers
- Wide data buses

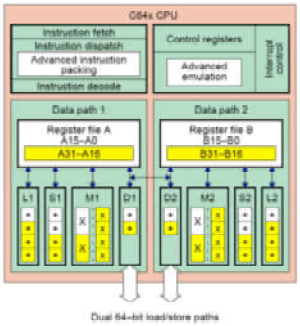
Green things are related to parallelism.

A C6415 DSP architecture



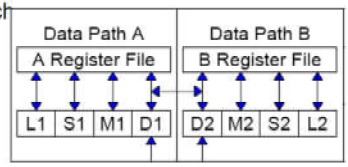
TMS320C64x

- 2 register banks (A, B)
  - Dual structure
  - 32-32-bit registers in each bank
- Register types:
  - Condition registers (3/register bank)
  - Address registers (4/register bank)
  - Data registers
- 8 Processing units: L, S, M, D (2 x 4) with different bandwidth
- Crossing data lines
- Special instructions

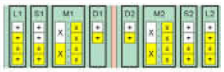


C64x processor cores: **Register banks**

- Two register banks (A,B)
  - Dual structure
- 32 32-bit registers in each bank
- Register types
  - Condition registers (3 /register bank)
  - Adress registers (4 /register bank)
  - Data registers



C64x Processing units



M1, M2 egységek
Szorzó összeadó (MAC) egység
16 x 32 multiply operations Dual and quad multiply operations Dual and quad multiply with add/subtract operations (MAC) Rotation Variable shift operations

S1, S2 egységek
Aritmetika, komaprálások, elágazások
32-bit arithmetic and logical operations 32/40-bit shifts Branches Dual and quad compare operations Dual and quad saturated arithmetic operations

L1, L2 egységek
Logikai és aritmetikai egység
32/40-bit arithmetic and compare ops 32-bit logical operations Byte shifts Dual /quad arithmetic operations Dual /quad min/max operations Quad 8-bit subtract with absolute value

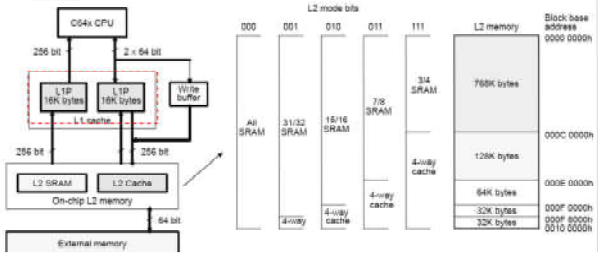
D1, D2 egységek
Címaritmetikai egység
32-bit add, subtract, linear and circular address calculation Loads and stores single and double words with constant offset 32-bit logical operations Dual 16-bit arithmetic operation

47

C64x: L1, L2 Cache-memory

L1 cache (2x16K): independent data and program caches

L2 cache (1024K): confugurable



Processing units

- M1, M2: multiplier and adder units (MAC)
- S1, S2: arithmetic, comparison and branches
- L1, L2: logic and arithmetic units
- D1, D2: Address arithmetic units

Instruction bus

- The instruction width of each of the 8 processing units is 32 bits -> 256-bit instruction bus width
  - It is a single bus, because the processors cannot branch independently!
  - Typically, 2-3 processors are working in the same time.

DMA (Direct Memory Access)

- DSP can access all I/O request through DMA
- 64 DMA channel
- Altogether 2 Gbyte/sec
- Linked DMA channel: Assigned to a specific event, if the event is occurred the DMA start automatically
- Chaining DMA channel: DMA channel starts at end of another DMA transfer (loop channel)
- 1D-2D channels

Peripherals

- Three parallel bus: 64 bits and 16 bits
- 3-3 memory controller
  - SDRAM controller
  - Synchronous controller
  - Asynchronous controller
- HPI or PCI interface
- Programable I/O channel (GPIO)
- Three serial port

Four independent DMA queue

- Concurrency (4 independent – concurrent row)
- Channels on different queues can interleave transfers on a cycle by cycle basis. e.g.:
  - cycle 1 queue 0
    - service a L2 cache miss to EMIFA (64 bit)
  - cycle 2 queue 1
    - Move data from a serial port to EMIFB (16 bit)
  - cycle 3 queue 3
    - the PCI/HPI could transfer data to mapped internal memory.
  - cycle 4 queue 4
    - the EMIFA could move data to a serial port (RS232)

7. Specify the time gain in pipeline operation. (1)
8. Define the bounds of pipeline operation. (2)

Speedup (S) of the pipelined processor over non-pipelined processor, when M tasks are executed:

$$\text{Speedup} = \frac{\text{Time for M instruction without pipelining}}{\text{Time for M Instruction with Pipelining}}$$

$$\text{Speedup} = M*T_p/(N+M-1)(T_s+T_{REG}) = T_p/(1+(N-1)/M)(T_s+T_{REG})$$

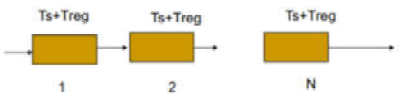
Where:

- T_p is the time of the operation without pipelining
- M is how many times we want to run our operation
- N is the number of sub-operations that we created from the original operation
- T_s is the sub-operation's runtime
- T_{reg} is an additional time due to data movements

If we want to use this operation many-many times, we end up with a big M, therefore:

$$\text{If } M>>>1 \quad \text{Speedup} = T_p/(T_s+T_{REG})$$

8. Define the bounds of pipeline operation. (2)

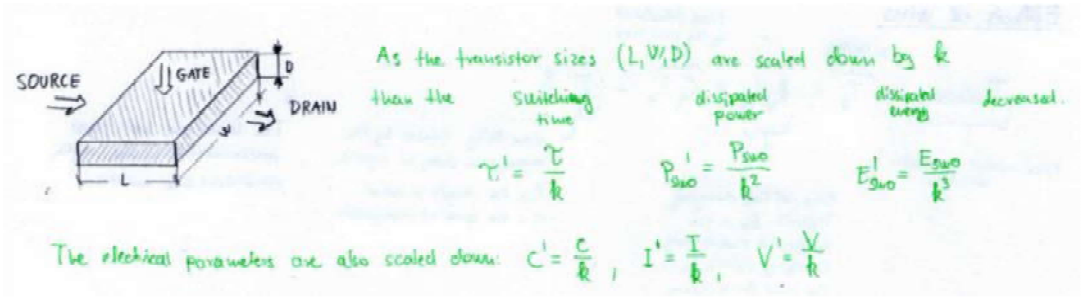


- The time of a single instruction do not reduce by using pipelining
- T_S is the time of a single operation, but the data transfer/storage time must be added (T_{reg})
- The maximum T_S is one result per clock cycle
- Maximal operation time in pipeline processing is determined by the slowest stage
- the result is more realistic speedup without registers -> speedup can be infinite.

8. Define the bounds of pipeline operation. (2)
9. Effect of the Scaling Down to gate capacity and to the energy requirement (Figure+ explanation) (1)

Scaling down

- The computational power of integrated electrical units will be better and better by decreasing their size.



explanation) (1)

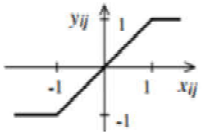
10. Give the CNN modell with the basic equations. (define the varables and the parameters), how can you define an elementary istruction? (2)

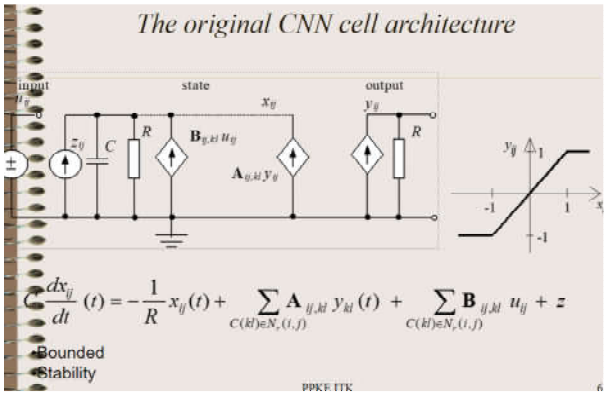
Cellular Neural/Nonlinear Networks (CNN)

- Identical analog processing elements called CNN nucleus
- Locally connected

The CNN State Equation

- Original CNN equation: $\dot{x}_{ij}(t) = \sum_{C(kl)\in N_r(i,j)} A_{ij,kl}\, y_{kl}(t) + \sum_{C(kl)\in N_r(i,j)} B_{ij,kl}\, u_{kl} + z_{ij}$
- Discretized CNN equation: $x_{ij}(k+1) = (1-h)x_{ij}(k) + h\left(\sum_{C(kl)\in N_r(i,j)} \mathbf{A}_{ij,kl}\, y_{kl}(k) + \sum_{C(kl)\in N_r(i,j)} \mathbf{B}_{ij,kl}\, u_{kl} + z_{ij} \right)$
- u_{kl}: input variable
- X_{ij}: state variable
- Y_{kl}: output variable
- A_{ij}, B_{ij}: template matrices





$$C\frac{dx_{ij}(t)}{dt} = -\frac{1}{R}x_{ij}(t) + \sum_{C(kl)\in N_r(i,j)} A_{ij,kl}\, y_{kl}(t) + \sum_{C(kl)\in N_r(i,j)} B_{ij,kl}\, u_{kl} + z_{ij}$$

IDK why is this written in this over-complicated form...

$$C\,\dot{x}_{ij}(t) = -\frac{1}{R}x_{ij}(t) + \sum_{C(kl)\in N_r(i,j)} A_{ij,kl}\, y_{kl}(t) + \sum_{C(kl)\in N_r(i,j)} B_{ij,kl}\, u_{kl} + z_{ij}$$

where: x_{ij} : state
y_{ij} : output
z_{ij} : bias
u_{ij} : input

$N_r(i,j)$ neighborhood around (i,j) with radius r
R: ?

$A_{ij,kl}$ $B_{ij,kl}$ C ?

$A = \begin{bmatrix} & \end{bmatrix}$ $B = \begin{bmatrix} & \end{bmatrix}$

Discretized CNN equation

parameters), how can you define an elementary istruction? (2)
11. Give the implicit integration form and specify the adventages and disadventages of the method. (2)

?

12. Give the discretized CNN state eq of the CASTLE architecture. (2)

Discretized CNN equation

$$x_{ij}(k+1) = (1-h)x_{ij}(k) + h\left[\sum_{C(kl)\in N_r(i,j)} A_{ij,kl}\, y_{kl}(k) + \sum_{C(kl)\in N_r(i,j)} B_{ij,kl}\, u_{kl} + z_{ij}\right]$$

Discretized CNN equation in full-range model:

$$x_{ij}(k+1) = \begin{cases} 1 & x_{ij}(k) > 1 \\ x_{ij}(k) & |x_{ij}(k)| \le 1 \\ -1 & x_{ij}(k) < -1 \end{cases}$$

$$x_{ij}(k) = (1-h)x_{ij}(k) + h\left[\sum_{C(kl)\in N_r(i,j)} A_{ij,kl}\, x_{kl}(k) + \sum_{C(kl)\in N_r(i,j)} B_{ij,kl}\, u_{kl} + z_{ij}\right]$$

Using systolic array processor in the next matrix

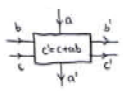
drawn on the exact structure and the flow of data and also how structured and how many processors should be used. (3) What kind of operation is performed in a single processor? (0: 5) How much of the surface and the time requirement? (0.5)

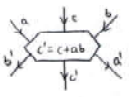
Systolic processor arrays

- arranged on a regular grid
- finite state machines with million states
- execute algebraic operations

Typical tasks: matrix operations
FFT, DFT
PDE with finite elements
Simulation

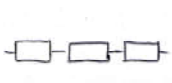
Processor

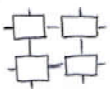


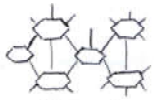


1) a,b,c values clocked into P_a, P_b, P_c
2) $P_c = P_c + P_a P_b$
3) P_a, P_b, P_c values clocked into a', b', c'

Layout


linear


orthogonal


hexagonal

There is a global clock and the time requirement of a single operation is one time unit.

2 Given a logic gate consisting circuit 02 Rent = constant. The chip wiring specific unit of capacity 5pF C / cm. CO 2 C 6-10'pF. If an 8 transistor (N fanout) driven by another average line length, it is many times the case is a waste of time for optimal alignment (if you have only the wire load)? (Dg = 20lum) (4)

the case is a waste of time for optimal alignment (if you have only the wire load)? (Dg = 20lum) (4)

3 Enter the symbolic computing model. It is correct to describe this model and why parallel computing. (3)

- Symbolic model

This is the software : the Turing machine ≡ grammar ≡ μ-recursive fuctions these three things are equivalent as they are used to describe an algorithm.

Symbolic model can be described as a graph of an algorithm (flowchart). It can also contain parallelized parts.

4 Enter the characteristic quantities of the complexity of a computer. (2)

5 Explain that for pipeline acceleration is not dependent on what the acceleration. (1)

6 Analyse the pipeline acceleration limits.

7 Describe (advantages, disadvantages) of floating gate-based FPGA programming mode.!

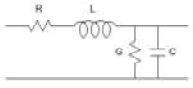
8 Specify that a modern FPGA. the type of wires occur, how to reduce the impact of wire delays (3p)

When the one dimensional distributed wire model have to be used in calculation of wire delay (2)

Where to use distributed parameter circuit model?

The distributed model have to be used if the rising time of a signal (t_r) smaller or comparable to the propagation time of the signal ower the transmission line (t_f).

$$t_f = \frac{l}{v}$$



- transmission line model should be used $t_r < 2.5t_f$
- concentrated parameter model can be used $t_r > 5t_f$

Between the two cases both models can be used

Critical length can be defined: $l_{krit} = \frac{vt_r}{2.5}$

Tipikus felfutási idők néhány IC technológiára

Technológia	On-Chip felfutási idők	Off-Chip felfutási idők
CMOS 0.5	2 nsec	4 nsec
Bipoláris	50 - 100 psec	200 - 400 psec
GaAs	20 - 100 psec	100 - 250 psec

Tipikus chip- és tokméretek

Alkatrész	Méretek
Die	1 cm
Chip carrier (single chip)	3 cm
Modul (multichip)	10 cm
PCB Board	50 cm

Kritikus vonalhosszak a felfutási idő függvényében (v=15cm/ns)

Felfutási idő tr (ps)	Kritikus hossz lkr=vtr/2.5 (cm)
50	0.3
100	0.6
250	1.5
500	4.5
1000	6.0
1500	9.0
8000	48.0

2. Define the relationship of the signal delays on PCBs and on ICs, and give the explanation as well. (2)

Jenn

3. Matrix vector multiplication by using systolic processor arrays draw the structure and the data stream as well. How many processors have to be used? Which kind of operaton a processor doing? (0.5) Define the ara and the time requiremnts of multiplication. (0.5)

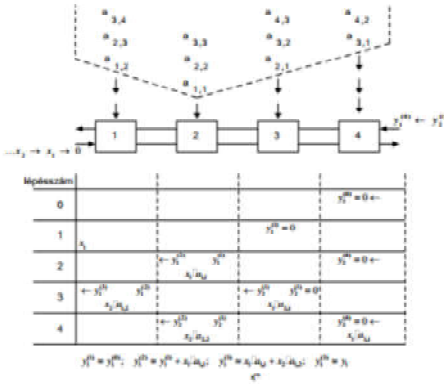
$$\begin{bmatrix} a_{11} & a_{12} & 0 & 0 & 0 \\ a_{21} & a_{22} & 0 & 0 & 0 \\ 0 & a_{32} & a_{33} & a_{34} & a_{35} \\ 0 & 0 & a_{43} & a_{44} & a_{45} \\ 0 & 0 & 0 & a_{54} & a_{55} \end{bmatrix} \cdot \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \end{bmatrix} = \begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \\ y_5 \end{bmatrix}$$

Let us have:

- One n x n band matrix, denoted by A, with bandwidth: w = p + q − 1,
- We want to compute: Ax = y

Systolic array solution:

- Processors are in linear arrangement
- Number of processors: w
- Problem is computed in a recursive way: $y_i^{(0)} = 0;\ y_i^{(k+1)} = y_i^{(k)} + a_{i,k} * x_k;\ y_i = y_i^{(n+1)}$
- $T = 2n + w\ (time);\ A = w\ (area)$ but if we use only one processor: $T = nw;\ A = 1$
- The processors are active in each second-time step only, but there is a shift in each time step.



How many processors have to be used?
Bandwidth: $\omega = p + q - 1 = 2 + 2 - 1 = 3$ → linear array of processors

Area: $A = \omega$
Time complexity: $T = 2n + \omega = 2 \cdot 5 + 3 = 10 + 5 = 15\ clk$
The processors are active in every second step of the iteration, but there is a time shift in each step.

A processor is doing addition. and multiplication?

4. Define the digital computation model. Explain why can it be used for parallel computation. (2)

- Digital model

$$x(k+1) = F(x(k), u(k+1)) \quad \text{next state}$$
$$y(k+1) = G(x(k+1), u(k+1)) \quad \text{output}$$

$x(k) \in D^k$ (electrical signals with discrete values)

Analog & Digital models can be parallelized: $y(t)$ does not depend on $y(t-1)$. Therefore the output can be calculated parallely. Also, if $x(k)$ is known, one can compute any $x(k+n)$ if $u(k+1), \dots, u(k+n)$ are given. ⇒ Can run parallely.

5. Define the parameters describing the computational complexity. (2)

Jenn

6. Describe a floating point DSP architecture, and define the function of the main blocks. Define the parts in the archetecture supporting the parallel operation. (4)

Ugyan az mint — fenn volt a TMS320C6x fix és lebegőpontos is.

7. Specify the time gain in pipeline operation. (2)

Speedup (S) of the pipelined processor over non-pipelined processor, when M tasks are executed:

$$\text{Speedup} = \frac{\text{Time for M instruction without pipelining}}{\text{Time for M Instruction with Pipelining}}$$

$$\text{Speedup} = M*T_p/(N+M-1)(T_s+T_{REG}) = T_p/[1+(N-1)/(M)](T_s+T_{REG})$$

Where:

- T_p is the time of the operation without pipelining
- M is how many times we want to run our operation
- N is the number of sub-operations that we created from the original operation
- T_s is the sub-operation's runtime
- T_{reg} is an additional time due to data movements

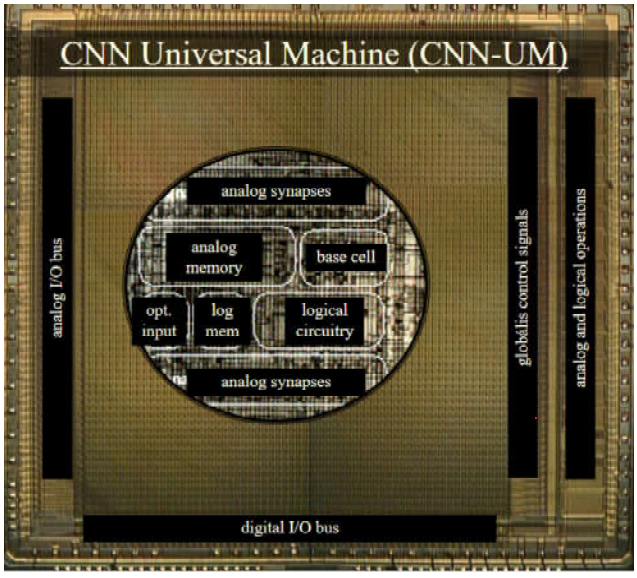
If we want to use this operation many-many times, we end up with a big M, therefore:

$$\text{If } M>>>1 \quad \text{Speedup} = T_p/(T_s+T_{REG})$$

9. Effect of the Scaling Down to dissipated power and to the speed (Figure+ explanation) (2)

Jobb ábrára minden van

10. Give the main blocks and the functions of a CNN-UM! (3)



Functions

Standard processing functions, like analog inversion or edge detection

Why is it universal?

These architecture can do everything what a turing machine can do. Which is a universal

hardware

Define the Negentropy (2)

$\Delta E_{dis} \geq \Delta S_L \cdot E_{SWO}$ ← if we want to compute a computational operator

Logikai entrópia (rendezetlenség) csökkenése = negentrópia — size of the potential states

$$\Delta S_L = \log_2 \frac{\text{lehetséges állapotok terének mérete (bit)}}{\text{eredmény mérete (bit)}} = \log_2 \frac{2^b p}{a}$$ ← size of the result.

b = input size
p = variable size
a = output size

ha $p = a \rightarrow \Delta S_L = b$, ha $a = 1 \rightarrow \Delta S_L = b \log_2 p$

E_{SWO}: elemi kapcsolási energia kell t idő alatt egy valódi elemi kapunál (inverter vagy 2bemenetű NAND kapu).
$\sim kT$ elméletileg szükséges energia kell, ezért az eszköz hatékonysága kT/E_{SWO}

ΔE_{dis} = dissipated energy

Ha nincs mellékműveletvégzé, akkor a számítási műveletek elvégzéséhez szükséges döntéssorozat elvégzéséhez legalább $\Delta E_{dis} = \Delta S_L * E_{SWO}$
Valójában: $\Delta E_{dis} <= \Delta S_L * E_{SWO}$

the used wires have to be changed (energy requirement)

Az adatokat el kell juttatni a döntést végző elemekhez:
1 bit esetén E_{SWO}, *p* bit esetén pE_{SWO} energia kell (ΔS_S=p).
A használt vezetékeket is fel kell tölteni: $(l/l_{SWE}) \cdot E_{SWO}$

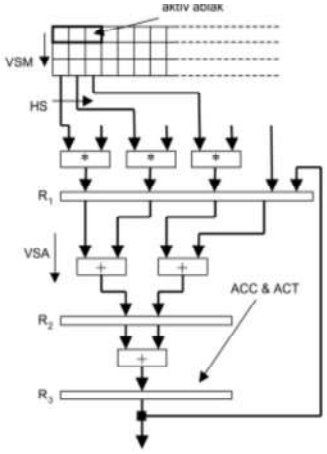
$\frac{l}{l_{SWE}} \cdot E_{SWO}$ | l - length of the wire | l_{SWE} : unit wire length

Az energia – entrópia hatásfoka tehát:
(A számítás véges felületen, időben, energiával stb történik)

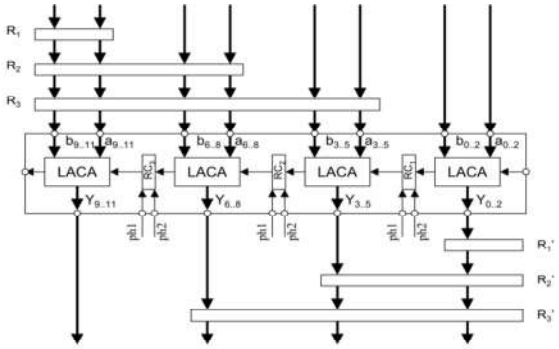
$$\eta = \frac{(\Delta S_L + \Delta S_S + \Delta l/l_{SWE}) \cdot E_{SWO}}{E_{Di}}$$

(energy efficiency)

2. Draw the pipelined arithmetic unit of the CASTLE. How can you estimate the size of the work area?(2)



CASTLE architecture with pipeline between the multipliers and adders.



Pipeline inside the multipliers and adders

Size of the work area?

Képfeldolgozásnál a CNN celláit egy-egy képpontnak feleltetjük meg, azaz minden cellára egy képpont értéket töltünk mind a bemenetre, mind az állapotra. Tehát egy *M* × *N*-es kép feldolgozásához
egy *M* × *N*-es négyzetrács struktúrájú CNN-re van szükség.

3. Adott egy logikai kapukból álló áramkör, ahol a chipen a vezetékek fajlagos egységnyi kapacitása C_u=1.7pF/cm. C_o=2 C_g=2•10^{-3}pF. Ha egy tranzisztor 10 (N FANOUT) másikat hajt meg átlagos hosszúságú vezetéken, akkor ez hányszoros időveszteséget jelent optimális illesztés esetén (ha csak a vezetéket tekintjük terhelésnek)? (dg=20µm) (4)

3. Adott egy logikai kapukból álló áramkör , és számolja ki azt a kritikus jelfelfutási időt (t_r), amely esetén az on-chip összeköttetéseket tápvonallal kell modellezni! (dg=2µm, v=16cm/ns) (4)