Parallel Computing With Water

Alec Henderson[1], Radu Nicolescu[1], Michael J. Dinneen[1], TN Chan[2], Hendrik Happe[3], and Thomas Hinze[3]

[1]School of Computer Science
The University of Auckland, Auckland, New Zealand

[2]Compucon New Zealand, Auckland, New Zealand

[3]Department of Bioinformatics
Friedrich Schiller University of Jena, Jena, Germany

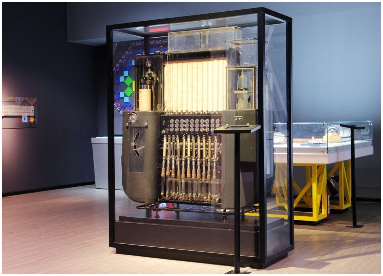
Outline of presentation

The power of water

Coromandel, New Zealand

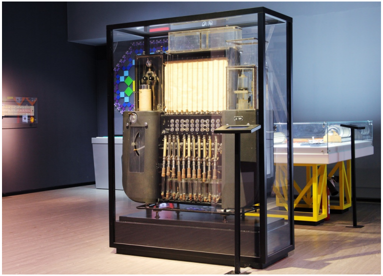
The magical sound,
of the cascading water,
natural beauty

Water integrator



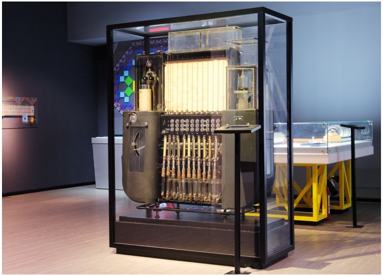
- First model built in 1936, in USSR; modular model in 1941, standard unified units in 1949-1955

Water integrator



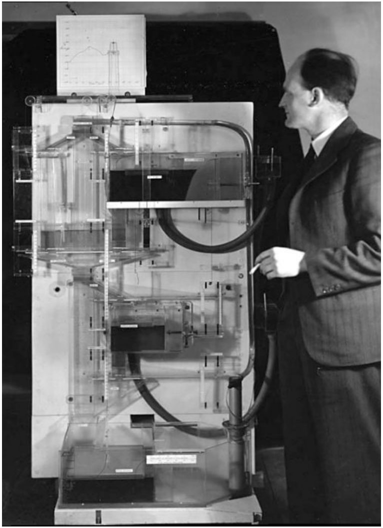
- First model built in 1936, in USSR; modular model in 1941, standard unified units in 1949-1955
- Used to solve inhomogeneous differential equations with applications such as: solving construction issues in the sands of Central Asia and in permafrost and in studying the temperature regime of the Antarctic ice sheet

Water integrator



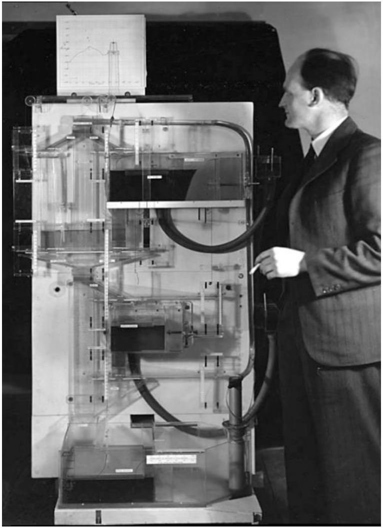
- First model built in 1936, in USSR; modular model in 1941, standard unified units in 1949-1955
- Used to solve inhomogeneous differential equations with applications such as: solving construction issues in the sands of Central Asia and in permafrost and in studying the temperature regime of the Antarctic ice sheet
- Only surpassed by digital computers in the 1980's.

MONIAC (Monetary National Income Analogue Computer)



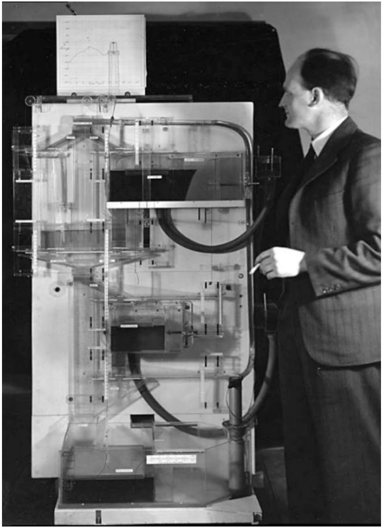
- MONIAC (Monetary National Income Analogue Computer) also known as the Phillips Hydraulic Computer and the Financephalograph

MONIAC (Monetary National Income Analogue Computer)



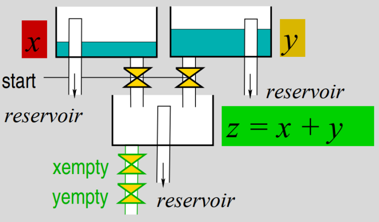
- MONIAC (Monetary National Income Analogue Computer) also known as the Phillips Hydraulic Computer and the Financephalograph
- First built in 1949 by New Zealand economist Bill Phillips to model the UK economy.

MONIAC (Monetary National Income Analogue Computer)



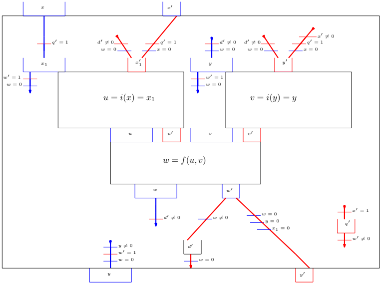
- MONIAC (Monetary National Income Analogue Computer) also known as the Phillips Hydraulic Computer and the Financephalograph
- First built in 1949 by New Zealand economist Bill Phillips to model the UK economy.
- Built as a teaching aid it was discovered that it was also an effective economic simulator.

Previous work[1]



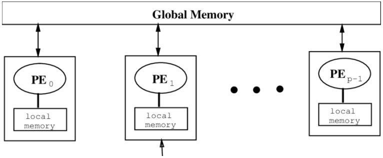
- No centre of control.
- Water flows if and only if all valves on a pipe are open.
- Water flows between tanks concurrently.

[1] Thomas Hinze et al. "Membrane computing with water". In: *J. Membr. Comput.* 2.2 (2020), pp. 121–136.



- Proved Turing complete via μ-recursive functions.
- Control tanks on input and output.

Problems



- Design a "practical" computational device.
- Show how to utilise the parallelism of the model via construction of Parallel Random Access Machine (PRAM).

RAM

Program p : GCD(a,b)

$r_3 \leftarrow r_1 \ominus r_2$	3	3	1	2
TRA 6 $r_3 > 0$	6	6	3	
$r_3 \leftarrow r_2 \ominus r_1$	3	3	2	1
TRA 8 $r_3 > 0$	6	8	3	
TRA 10 $r_1 > 0$	6	10	1	
$r_1 \leftarrow r_1 \ominus r_2$	3	1	1	2
TRA 1 $r_1 > 0$	6	1	1	
$r_2 \leftarrow r_2 \ominus r_1$	3	2	2	1
TRA 1 $r_2 > 0$	6	2	1	

- GCD(a,b): program of $m = 9$ lines
- Sequence of registers $r_1 = a$ (then result), $r_2 = b$, $r_3 = 0$

Op codes

1. $r_i \leftarrow C$: assign a constant value C to register i.
2. $r_i \leftarrow r_j \oplus r_k$: add the value of two registers j and k and assign to register i.
3. $r_i \leftarrow r_j \ominus r_k$: subtract from register j the value stored in k and assign to register i.
4. $r_i \leftarrow r_{r_j}$: get the value y from register j, then get the value from register y and assign to register i
5. $r_{r_i} \leftarrow r_j$: get the value y from register j, then get the value x from register i and assign y to register x.
6. TRA m $r_i > 0$: go to program line m (control transferred to line m of the program) if r_i greater than 0, otherwise go to the next line.

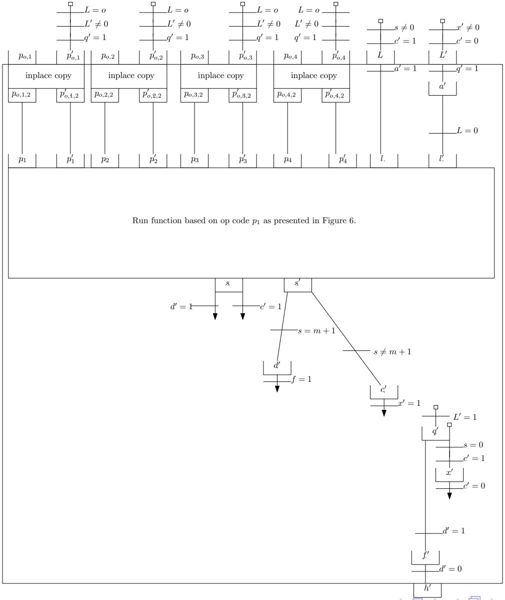
Op codes

Table: Operations and there corresponding opcodes.

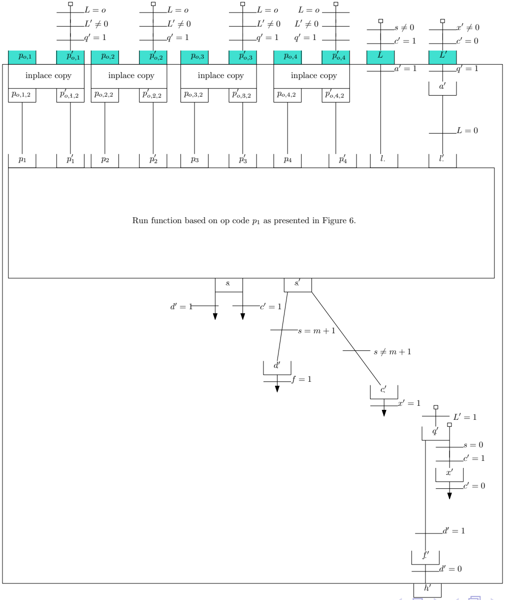
Name	Operation	Opcode
const	$r_i \leftarrow C$	$1\ i\ C$
add	$r_i \leftarrow r_j \oplus r_k$	$2\ i\ j\ k$
sub	$r_i \leftarrow r_j \ominus r_k$	$3\ i\ j\ k$
indr	$r_i \leftarrow r_{r_j}$	$4\ i\ j$
indw	$r_{r_i} \leftarrow r_j$	$5\ i\ j$
tra	TRA m $r_i > 0$	$6\ m\ i$

Outer loop

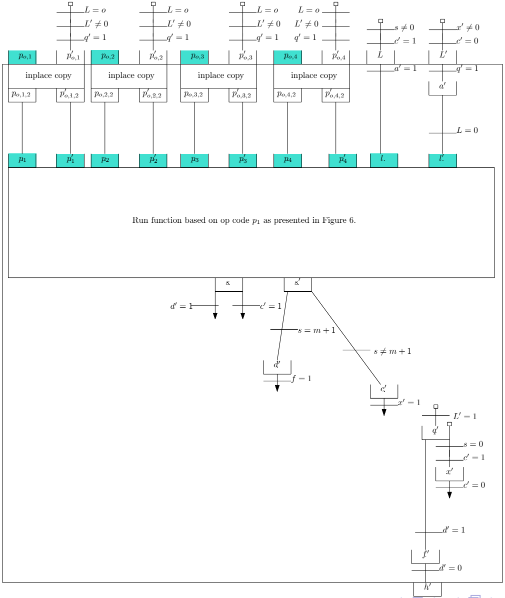
```
Procedure RAM(p, L)//p = [p_{1,1}, p_{1,2}, p_{1,3}, p_{1,4}, p_{2,1}, p_{2,2}, ..., p_{m,4}]
    p_1 ← p[4 ∗ (L − 1)]; p_2 ← p[4 ∗ (L − 1) + 1]
    p_3 ← p[4 ∗ (L − 1) + 2]; p_4 ← p[4 ∗ (L − 1) + 3]
    s ← run_op(p_1, p_2, p_3, p_4, L)
    if s ≠ m + 1 then
        RAM(p, s)
    halt
```


$L = o$
$L' \neq 0$
$q' = 1$
$s \neq 0$
$c' = 1$
$x' \neq 0$
$c' = 0$
$p_{o,1}$ $p'_{o,1}$ $p_{o,2}$ $p'_{o,2}$ $p_{o,3}$ $p'_{o,3}$ $p_{o,4}$ $p'_{o,4}$ L L'
inplace copy
$a' = 1$
$q' = 1$
a'
$p_{o,1,2}$ $p'_{o,1,2}$ $p_{o,2,2}$ $p'_{o,2,2}$ $p_{o,3,2}$ $p'_{o,3,2}$ $p_{o,4,2}$ $p'_{o,4,2}$
$L = 0$
p_1 p'_1 p_2 p'_2 p_3 p'_3 p_4 p'_4 $l.$ $l'.$
Run function based on op code p_1 as presented in Figure 6.
s s'
$d' = 1$
$c' = 1$
$s = m + 1$
$s \neq m + 1$
d'
$f = 1$
c'
$x' = 1$
$L' = 1$
q'
$s = 0$
$c' = 1$
x'
$c' = 0$
$d' = 1$
f'
$d' = 0$
h'

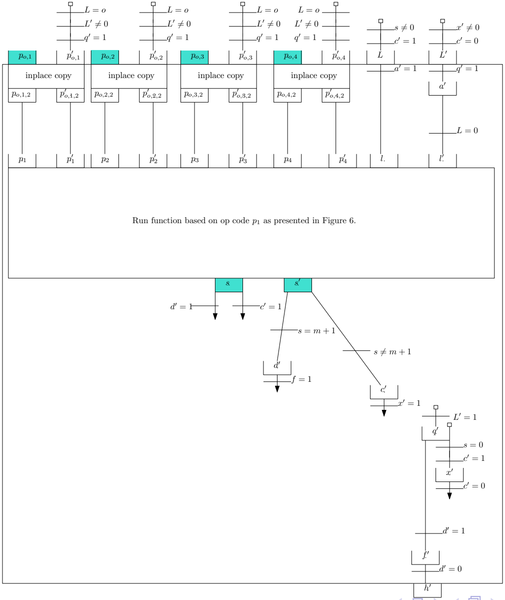
Outer loop Step 1



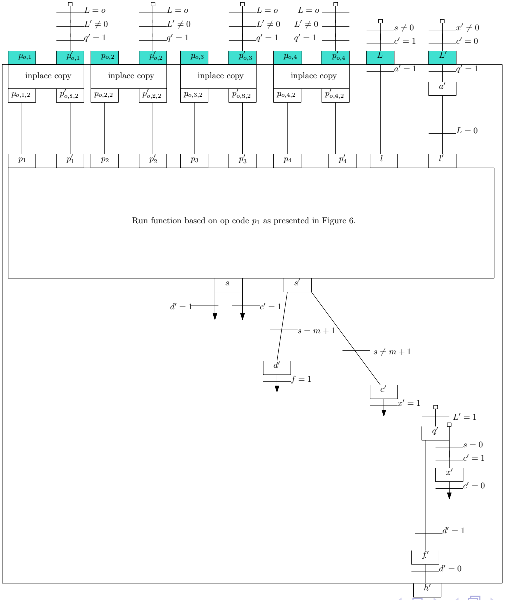
Outer loop Step 2

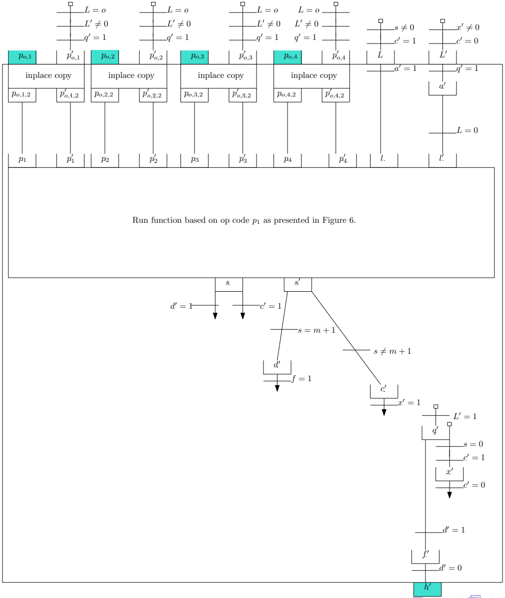


Outer loop Step 3



Outer loop Step 4(i)

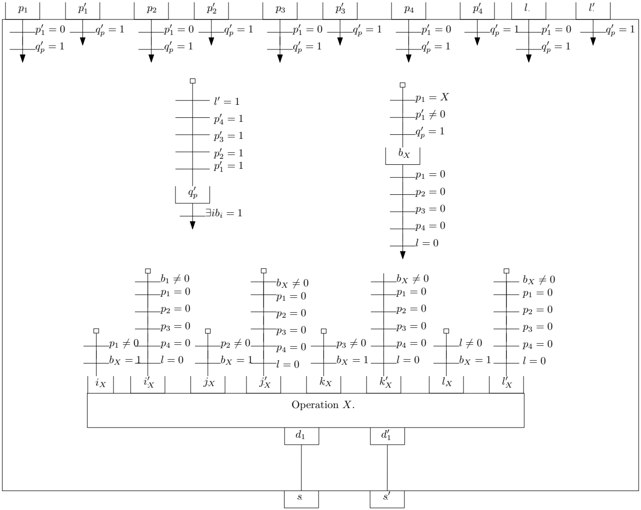



$L = o$
$L' \neq 0$
$q' = 1$
$p_{o,1}$
$p'_{o,1}$
$p_{o,2}$
$p'_{o,2}$
$p_{o,3}$
$p'_{o,3}$
$p_{o,4}$
$p'_{o,4}$
inplace copy
$p_{o,1,2}$
$p'_{o,1,2}$
$p_{o,2,2}$
$p'_{o,2,2}$
$p_{o,3,2}$
$p'_{o,3,2}$
$p_{o,4,2}$
$p'_{o,4,2}$
$s \neq 0$
$c' = 1$
L
$a' = 1$
$x' \neq 0$
$c' = 0$
L'
$q' = 1$
a'
$L = 0$
p_1
p'_1
p_2
p'_2
p_3
p'_3
p_4
p'_4
$l.$
$l'.$
Run function based on op code p_1 as presented in Figure 6.
s
s'
$d' = 1$
$c' = 1$
$s = m + 1$
$s \neq m + 1$
d'
$f = 1$
c'
$x' = 1$
$L' = 1$
q'
$s = 0$
$c' = 1$
x'
$c' = 0$
$d' = 1$
f'
$d' = 0$
h'

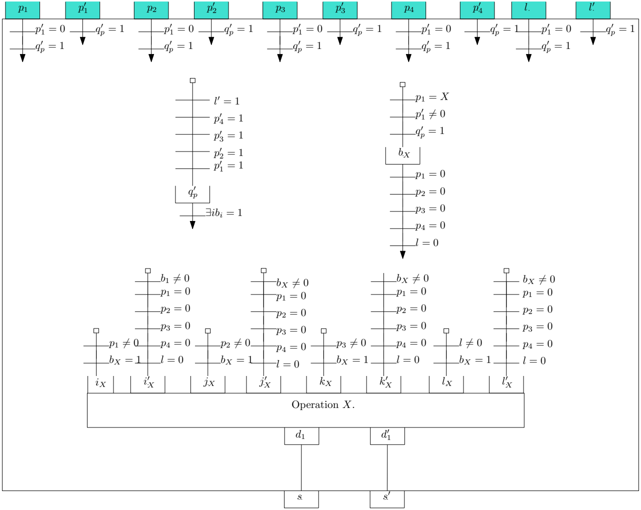
Operation selection

```
Procedure run_op(p1, p2, p3, p4, l)
    switch p1
        case 1
            return const(p2, p3, l)
        case 2
            return add(p2, p3, p4, l)
        case 3
            return sub(p2, p3, p4, l)
        case 4
            return indr(p2, p3, l)
        case 5
            return indw(p2, p3, l)
        case 6
            return tra(p2, p3, l)
```

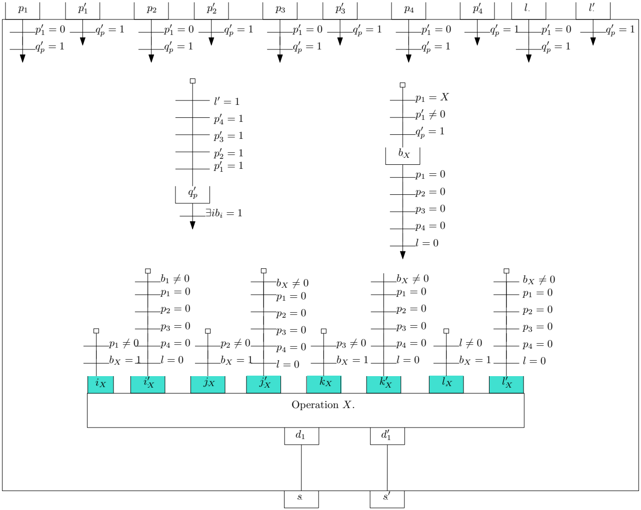
Operation selection



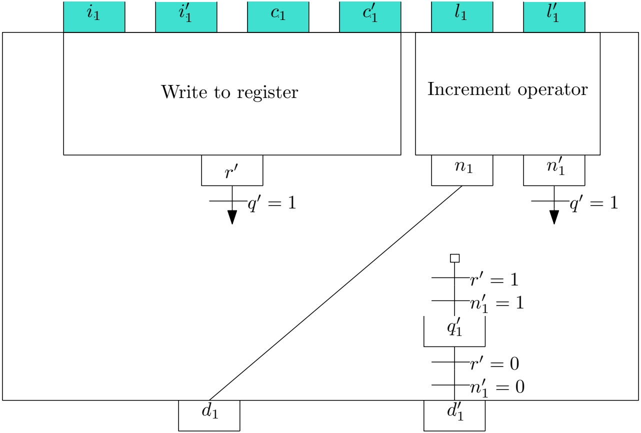
Operation selection step 1



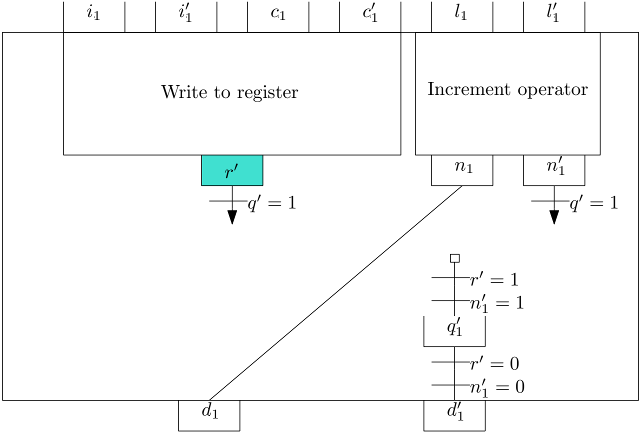
Operation selection step 2



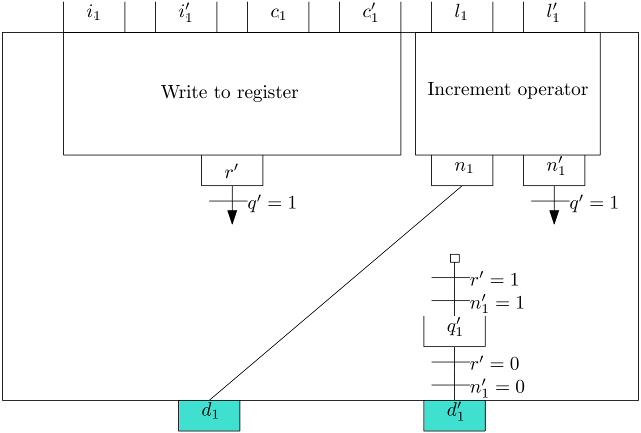
Constant step 1



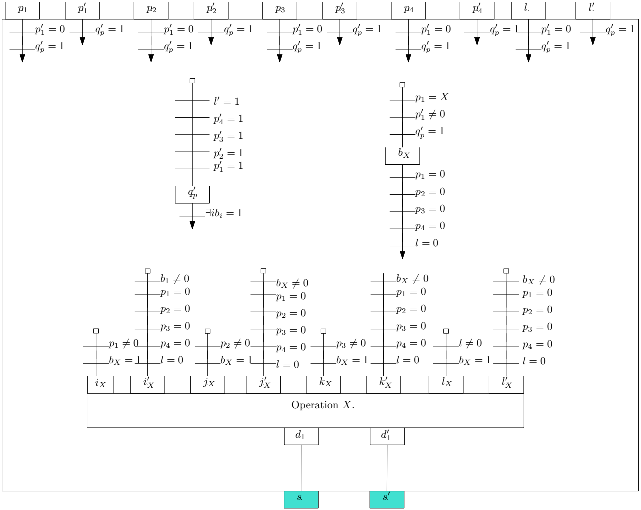
Constant step 2



Constant step 3



Operation selection step 3



Future Work

- Physical realisations.
- Equivalence to other P systems (cP, ?).
- Costs, e.g. pipes vs valves.

Questions

THANK YOU!

Any questions?