TERMA

Naval & Communication Systems

Aerospace

Space

Air defense

Radar Systems

Manufacturing Division

CISS
CENTER FOR INDLEJREDE SOFTWARE SYSTEMER

artist

Radar Systems Division

Surface Movement Radar Sensor Systems

VTS & Coastal Surveillance

Ship Radar Systems

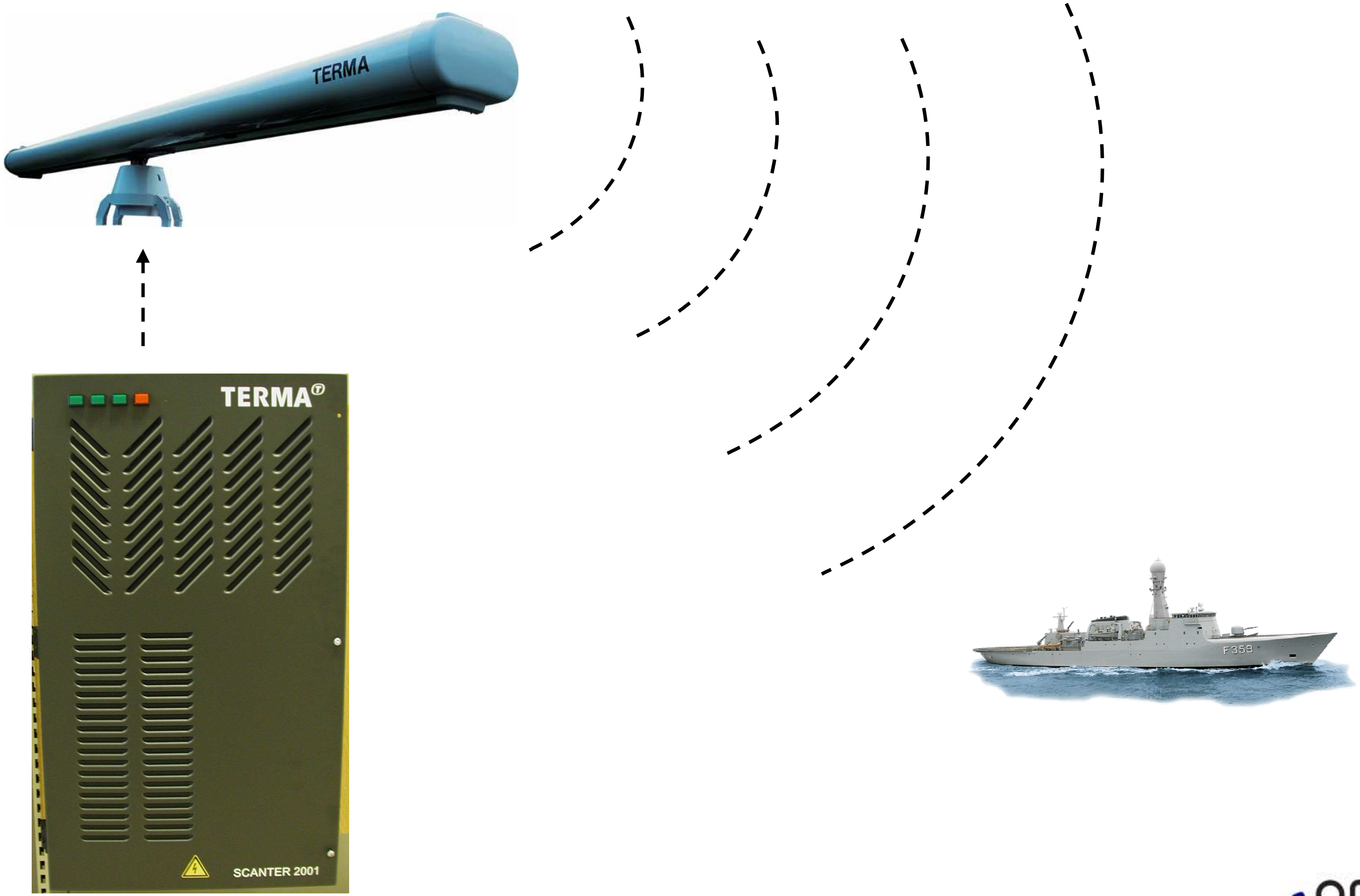
Airborne Surveillance Systems

Air Traffic Management

CISS
CENTER FOR INDLEJREDE SOFTWARE SYSTEMER

artist

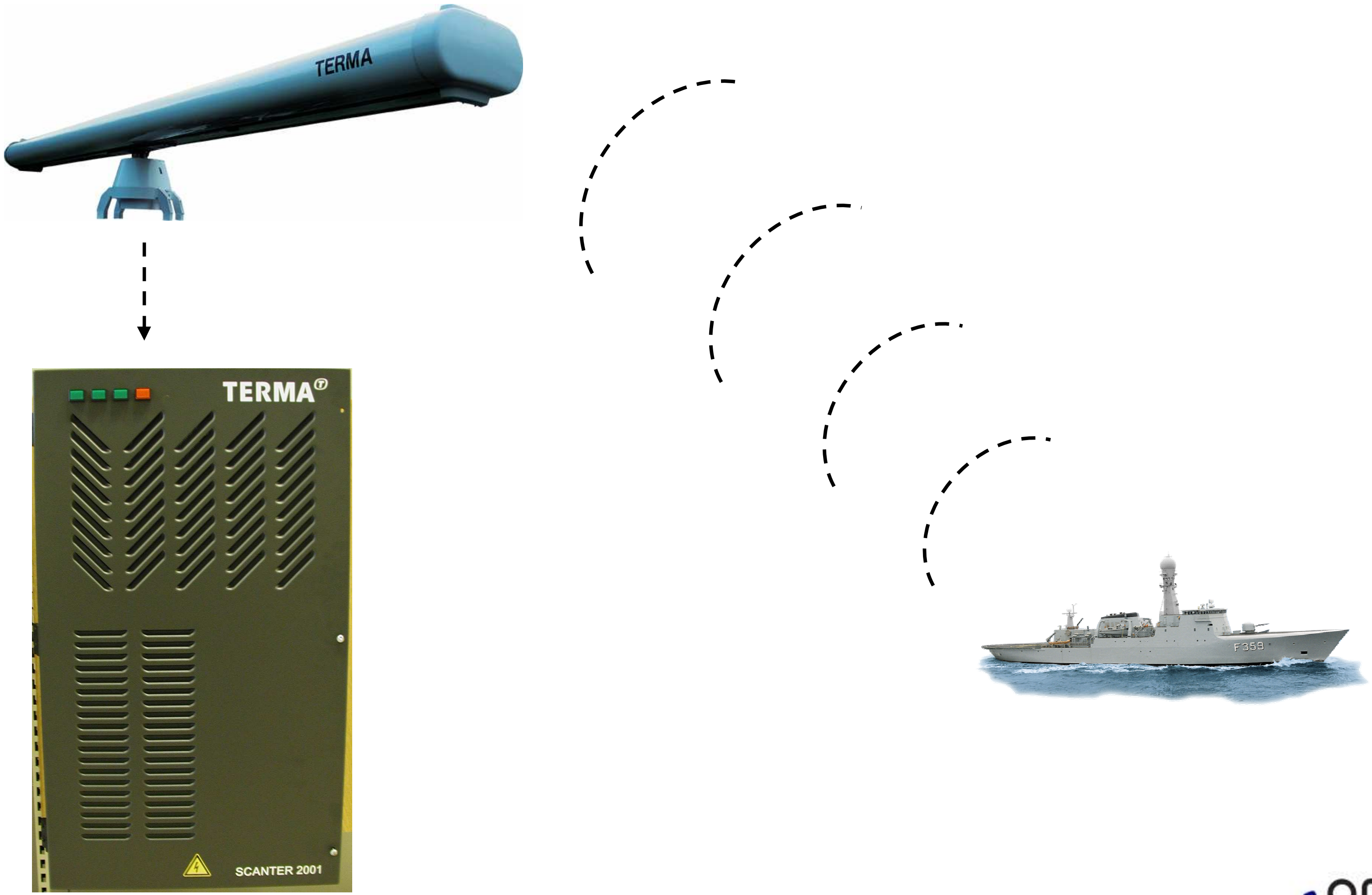
Radar Principles



CISS
CENTER FOR INDLEJREDE SOFTWARE SYSTEMER

artist

Radar Principles

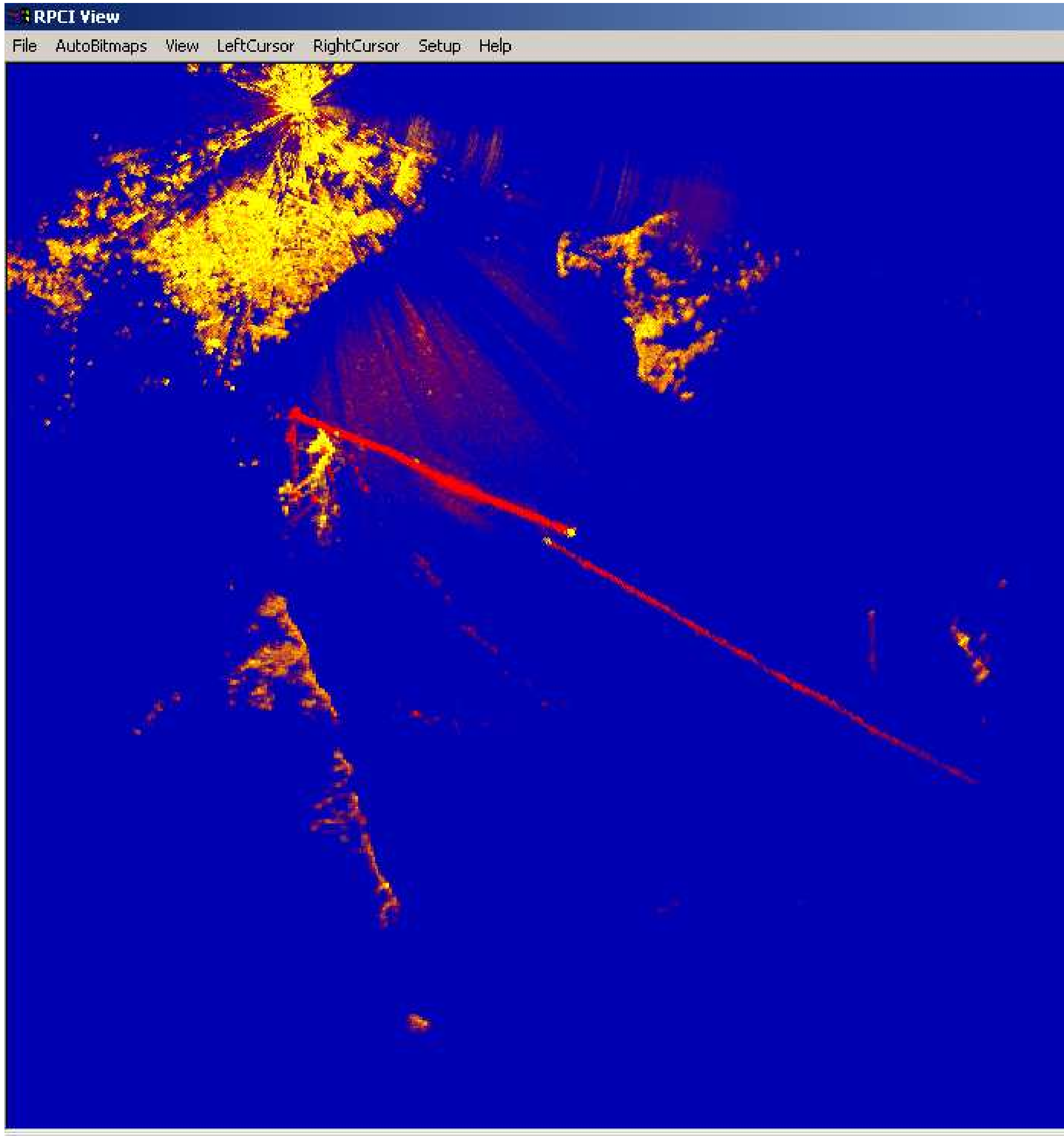


CISS
CENTER FOR INDLEJREDE SOFTWARE SYSTEMER

artist

Difficulties

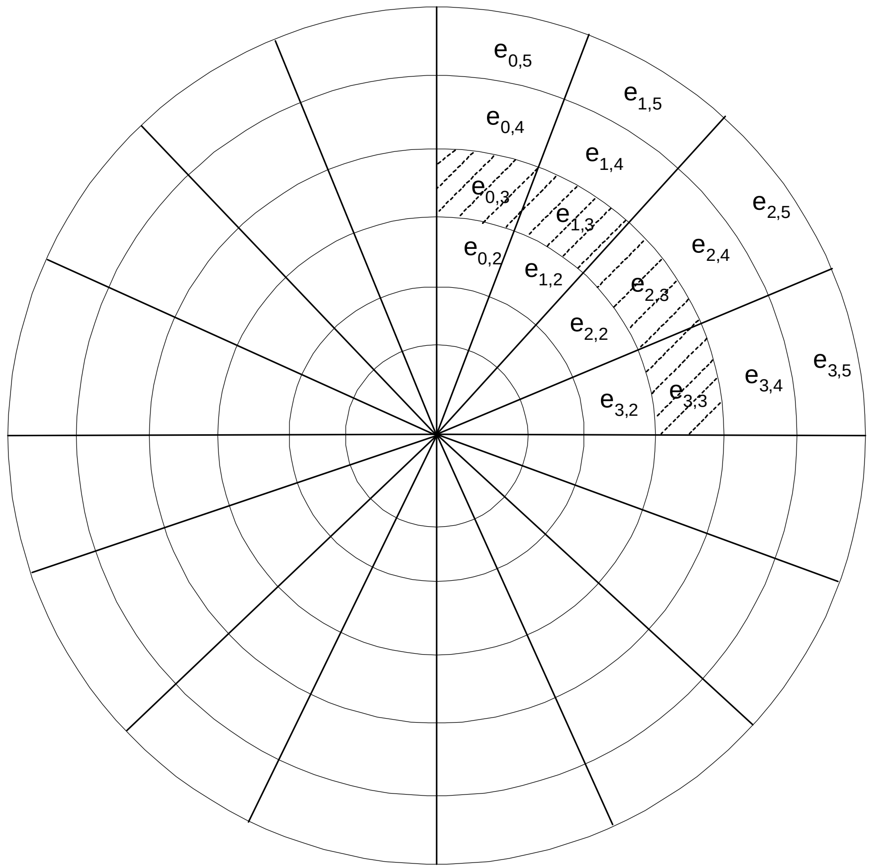
- Clutter
  - Rain
  - Sea
  - Ground / buildings
- Random noise
- Other radars



CISS
CENTER FOR INDLEJREDE SOFTWARE SYSTEMER

Sweep Integration

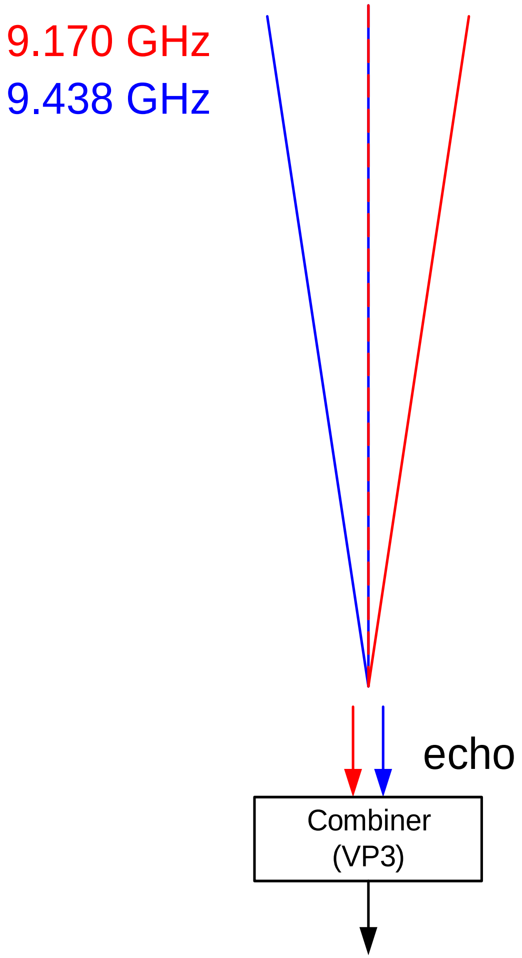
- Beam width of antenna, app. 0.4 deg



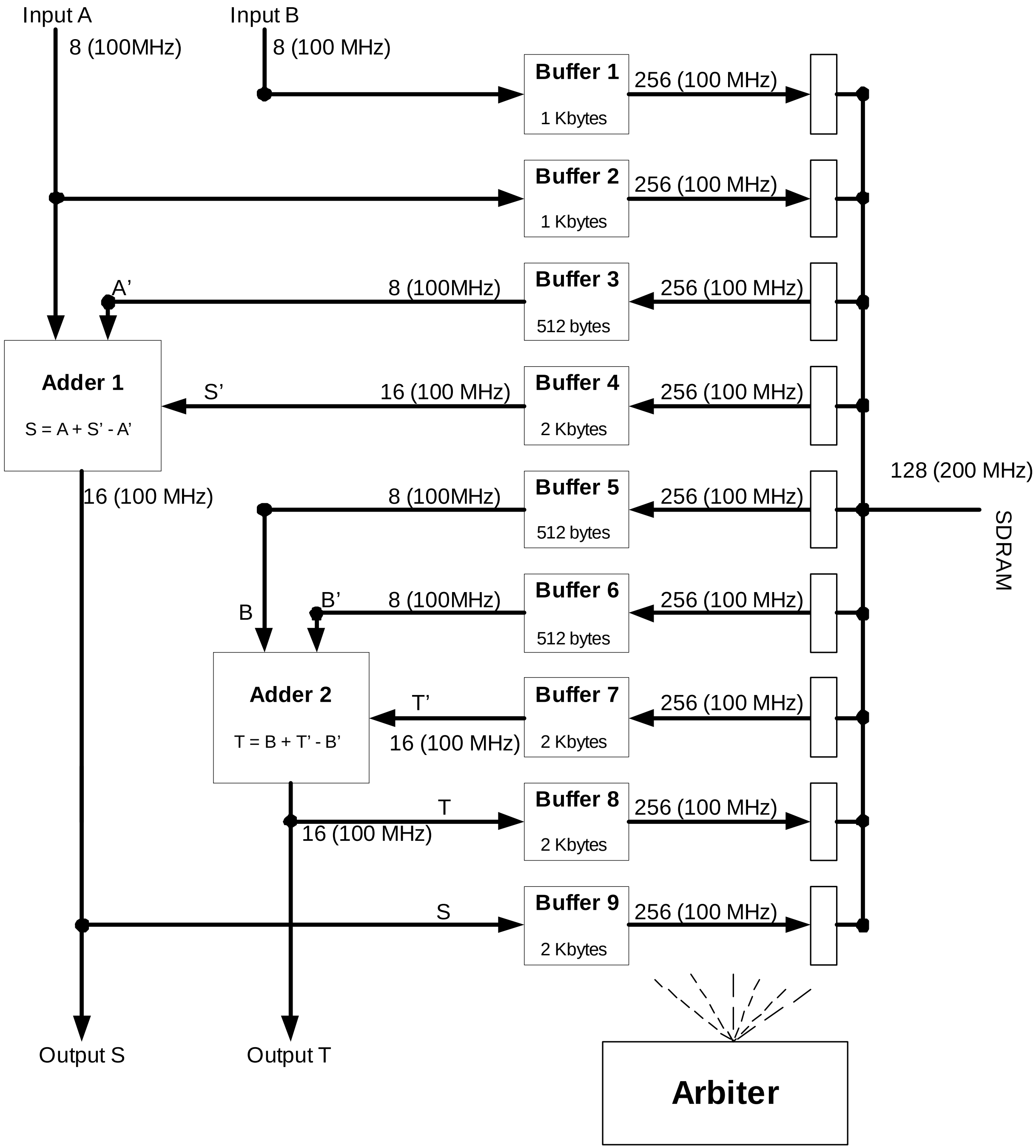
CISS
CENTER FOR INDLEJREDE SOFTWARE SYSTEMER

artist

Frequency Diversity



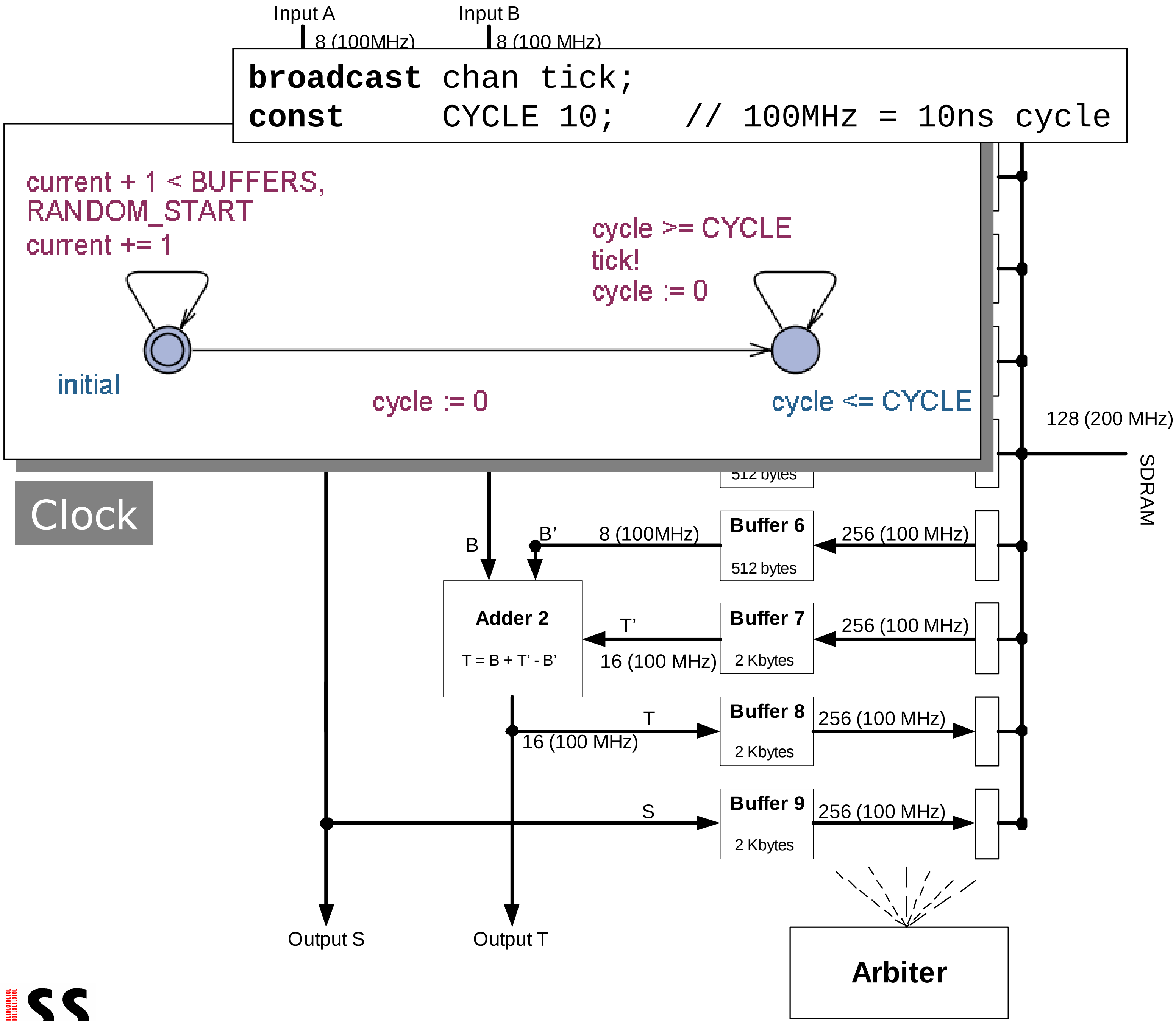
- Increased power
- Time delay between echos (clutter change over time)
- Different frequencies

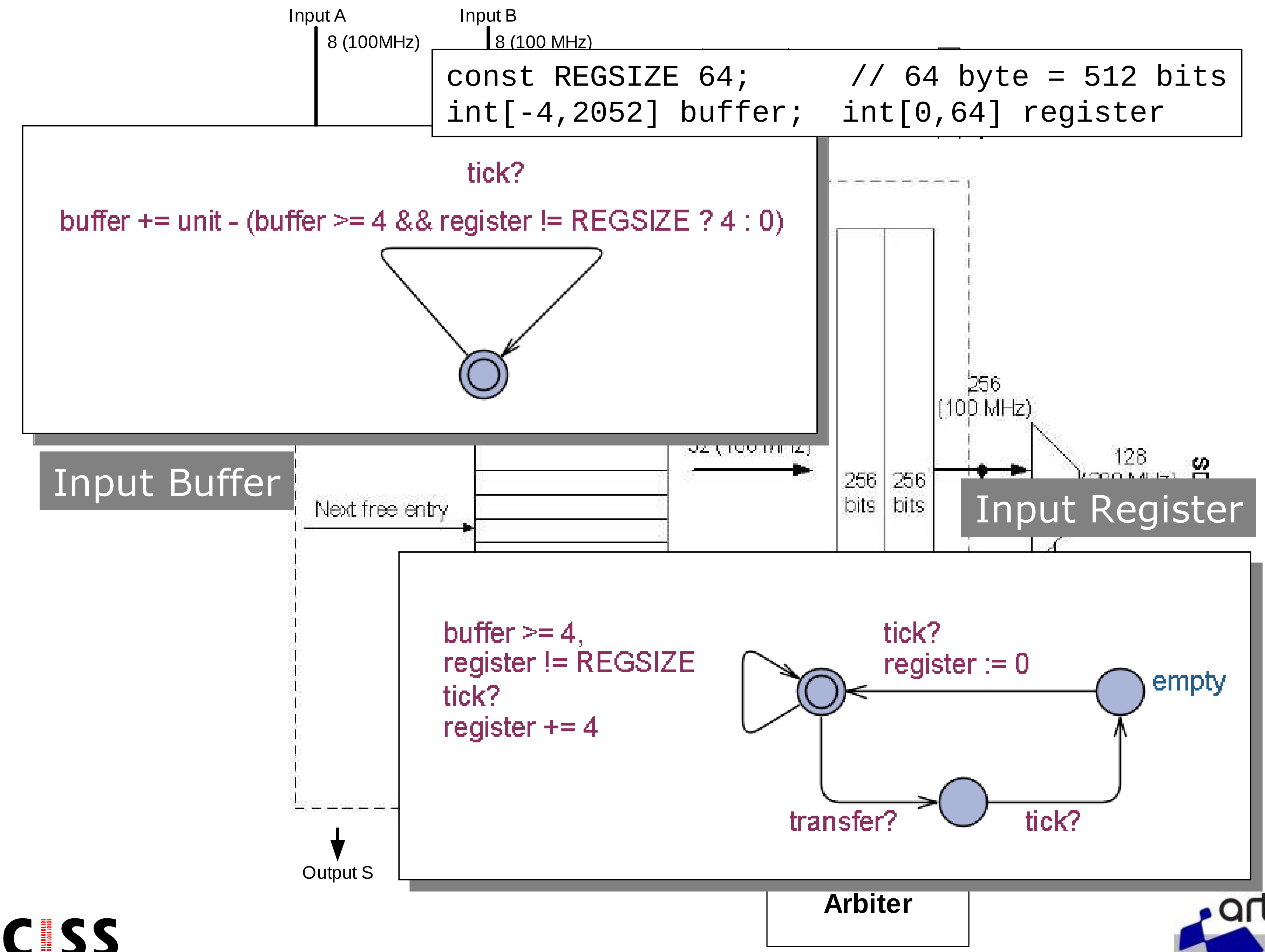


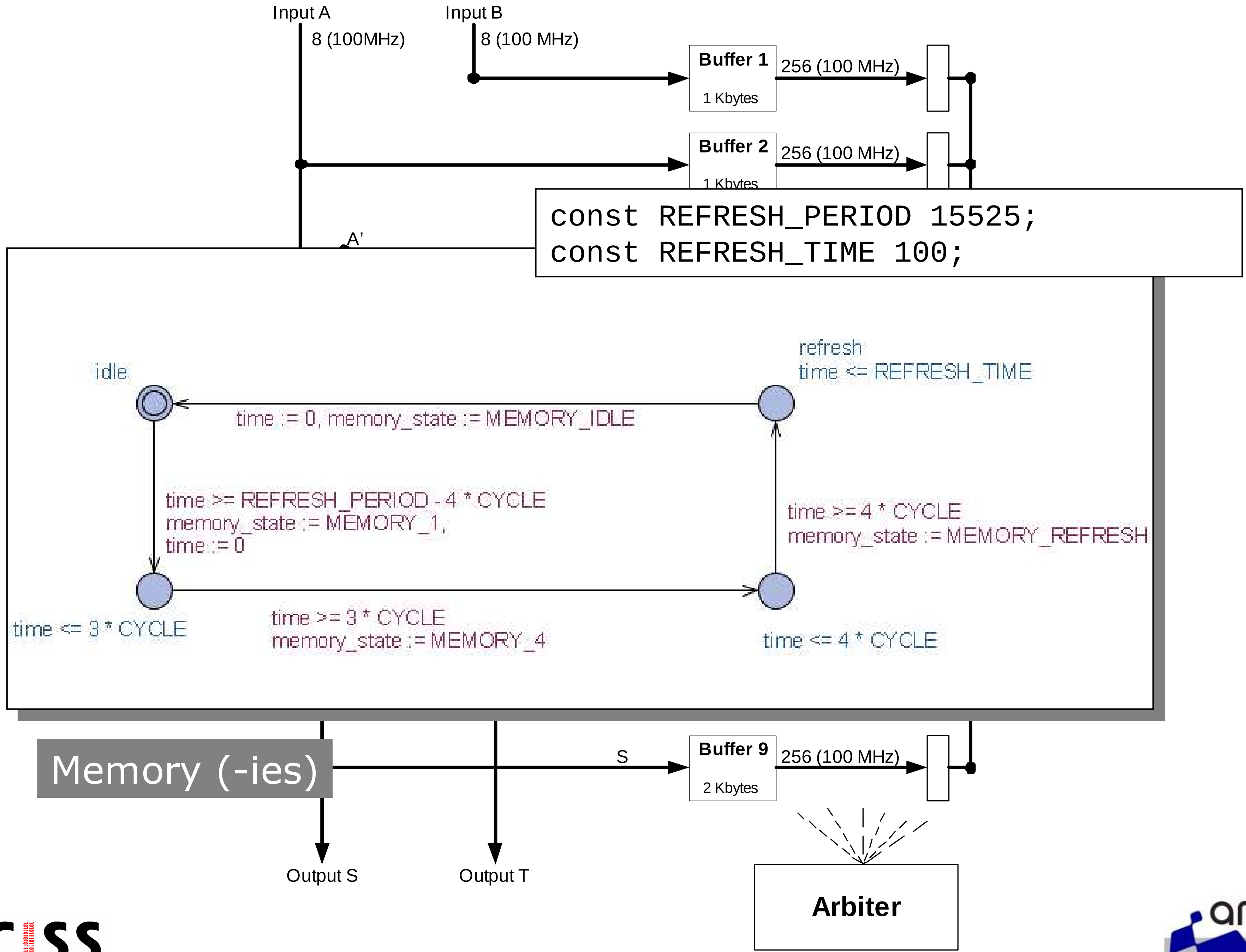
CISS
CENTER FOR INDLEJREDE SOFTWARE SYSTEMER

artist

```
broadcast chan tick;
const     CYCLE 10;    // 100MHz = 10ns cycle
```

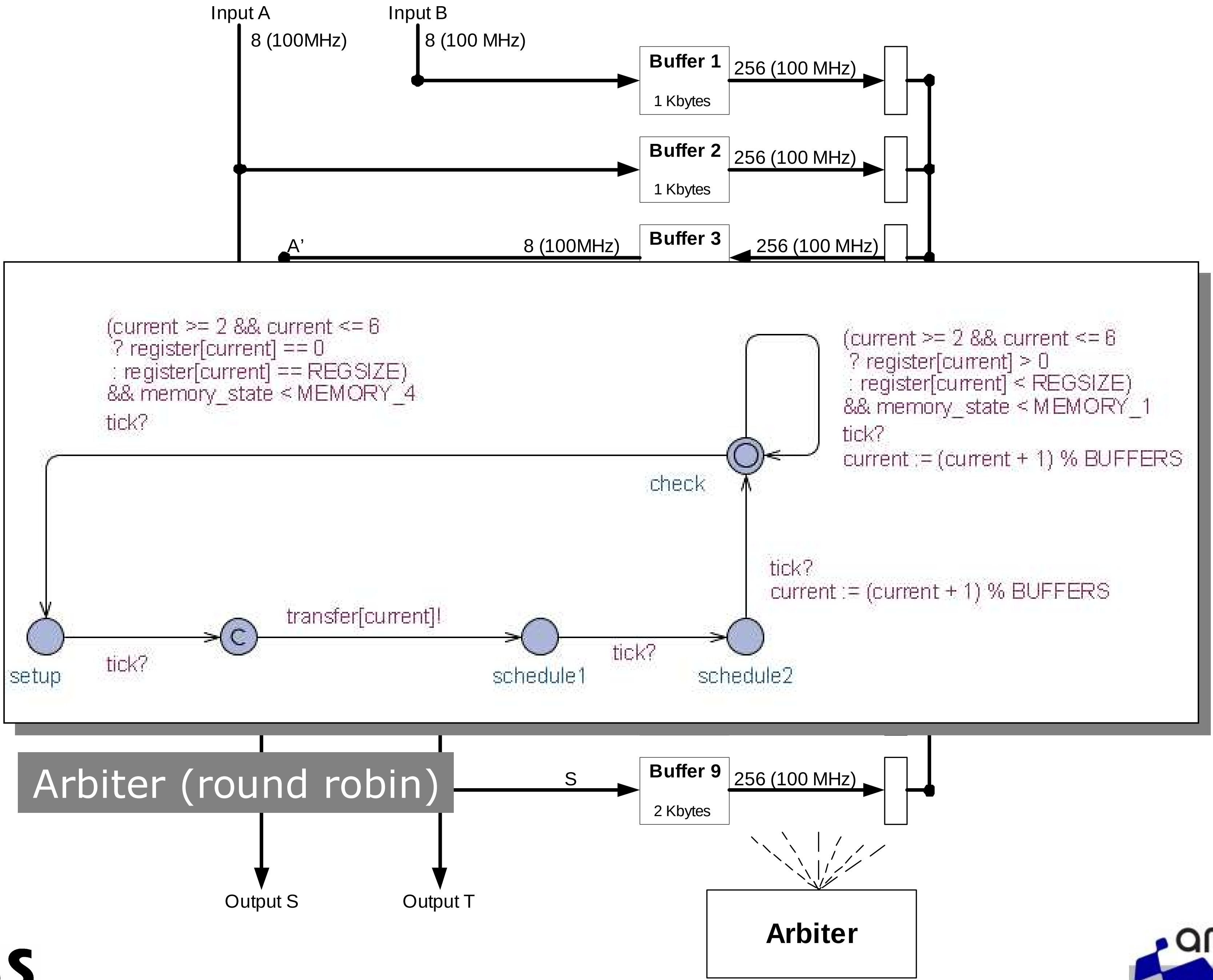



Input A
8 (100MHz)
Input B
8 (100 MHz)
const REGSIZE 64; // 64 byte = 512 bits
int[-4,2052] buffer; int[0,64] register
tick?
buffer += unit - (buffer >= 4 && register != REGSIZE ? 4 : 0)
Input Buffer
Next free entry
256 bits
256 bits
256
(100 MHz)
128
Input Register
buffer >= 4,
register != REGSIZE
tick?
register += 4
tick?
register := 0
empty
transfer?
tick?
Output S
Arbiter

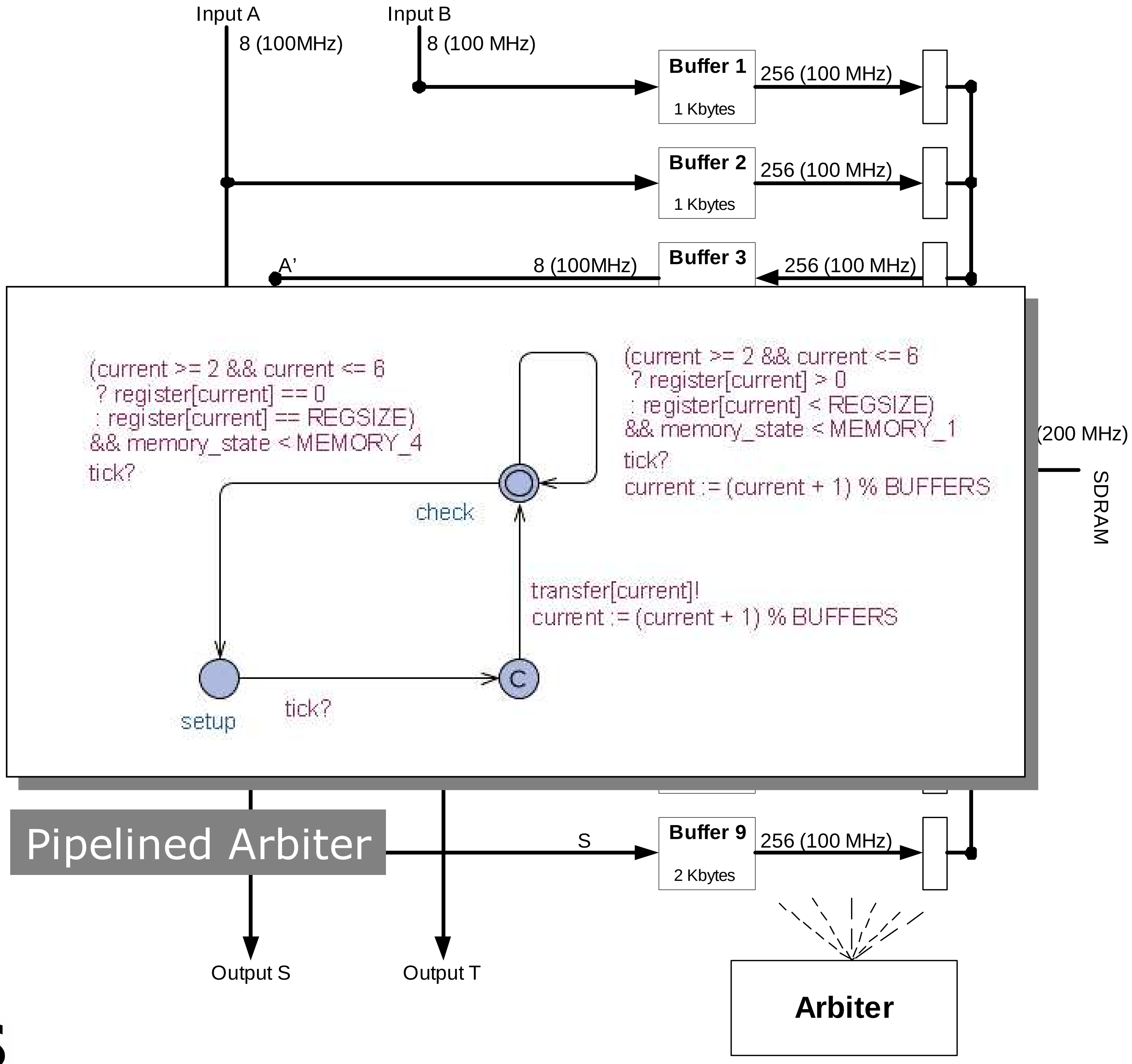


CISS
CENTER FOR INDLEJREDE SOFTWARE SYSTEMER

artist


Input A
8 (100MHz)
Input B
8 (100 MHz)
Buffer 1
1 Kbytes
256 (100 MHz)
Buffer 2
1 Kbytes
256 (100 MHz)
A'
8 (100MHz)
Buffer 3
256 (100 MHz)
(current >= 2 && current <= 6
? register[current] == 0
: register[current] == REGSIZE)
&& memory_state < MEMORY_4
tick?
(current >= 2 && current <= 6
? register[current] > 0
: register[current] < REGSIZE)
&& memory_state < MEMORY_1
tick?
current := (current + 1) % BUFFERS
check
tick?
current := (current + 1) % BUFFERS
transfer[current]!
C
setup
tick?
schedule1
tick?
schedule2
Arbiter (round robin)
S
Buffer 9
2 Kbytes
256 (100 MHz)
Output S
Output T
Arbiter

Pipelined Arbiter



Conclusions

- Final UPPAAL model 21 timed automata and 20 integer variables.
- UPPAAL verifies (< 1 min) that the used “round-robin” scheduler is CORRECT in the sense that no buffer over-flows or under-flows.
- Synthesis of scheduler optimal in terms of minum buffer size (6 bytes) (Juhan Ernits / Gera Weiss)
- Manual translation of VHDL descriptions to UPPAAL -- c(sh)ould be automated.

Much more to come …

DaNES

Danish Network for Intelligent Embedded Systems

CISS
CENTER FOR INDLEJREDE SOFTWARE SYSTEMER

artist

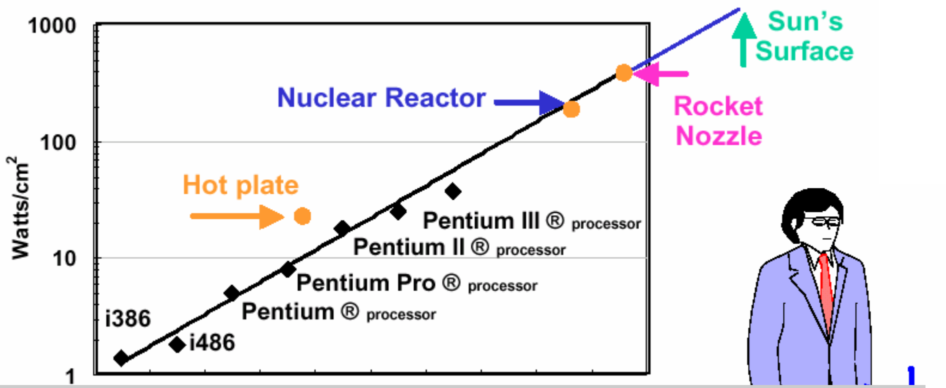
Dynamic Voltage Scaling

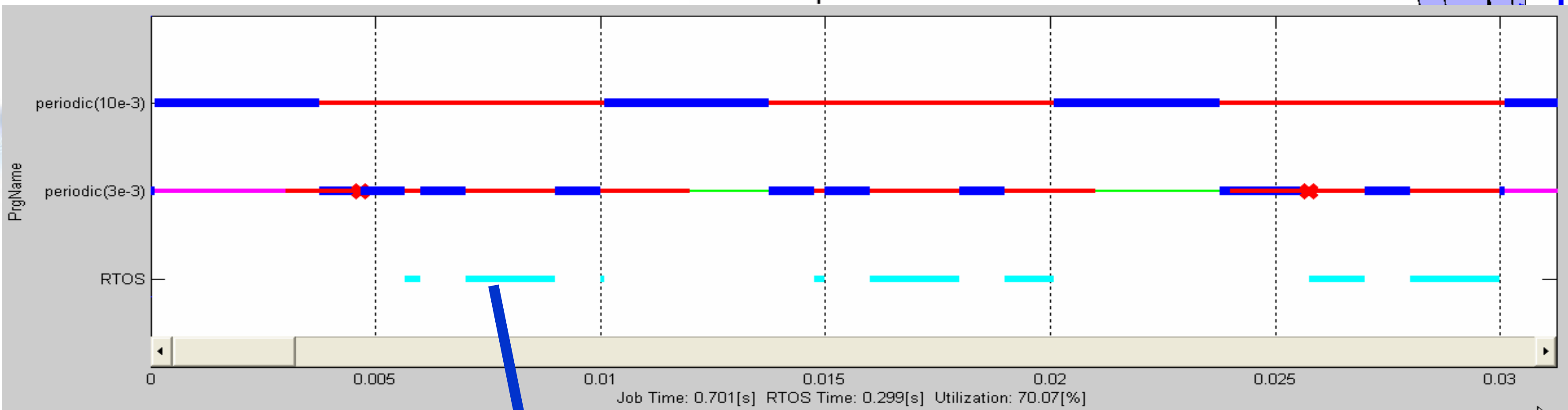
ANALOG DEVICES

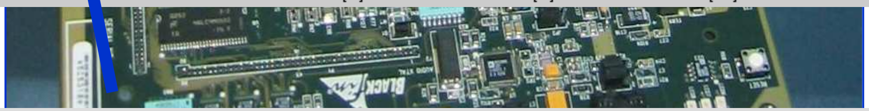
CISS
CENTER FOR INDLEJREDE SOFTWARE SYSTEMER

Power Management

Dynamic Voltage Scaling







Dynamically lower voltage supply (frequency) to utilize free CPU time

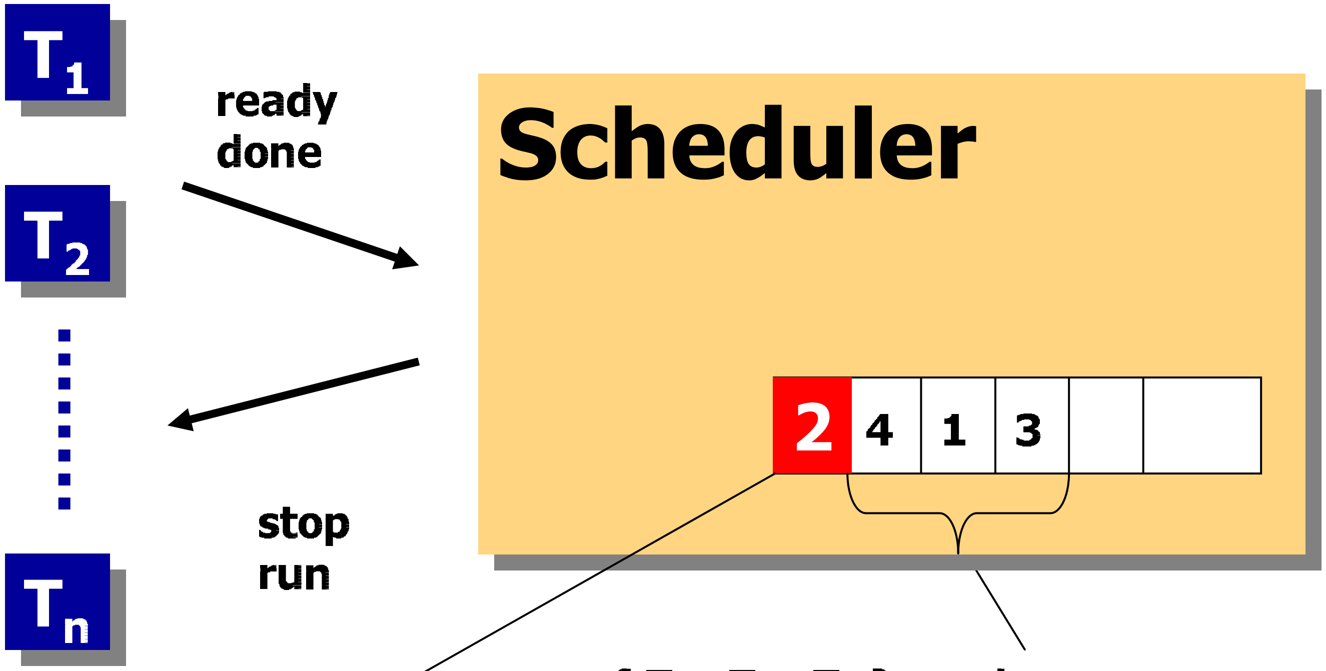
CISS
CENTER FOR INDLEJREDE SOFTWARE SYSTEMER

artist

Task Scheduling

utilization of CPU

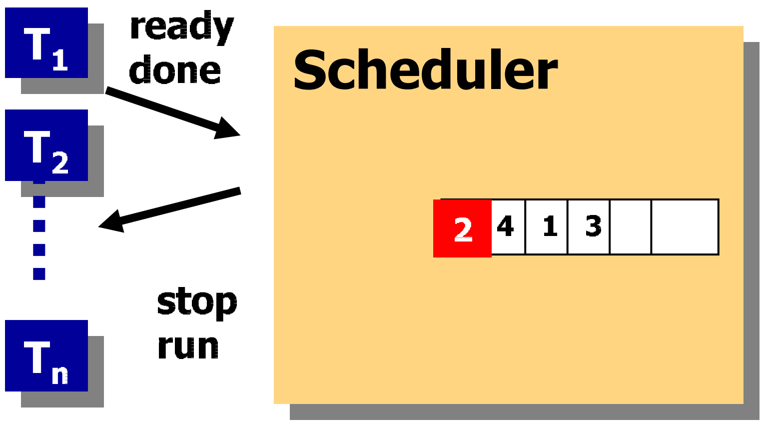
P(i), [E(i), L(i)], .. : period or earliest/latest arrival or .. for T_i
C(i): execution time for T_i
D(i): deadline for T_i

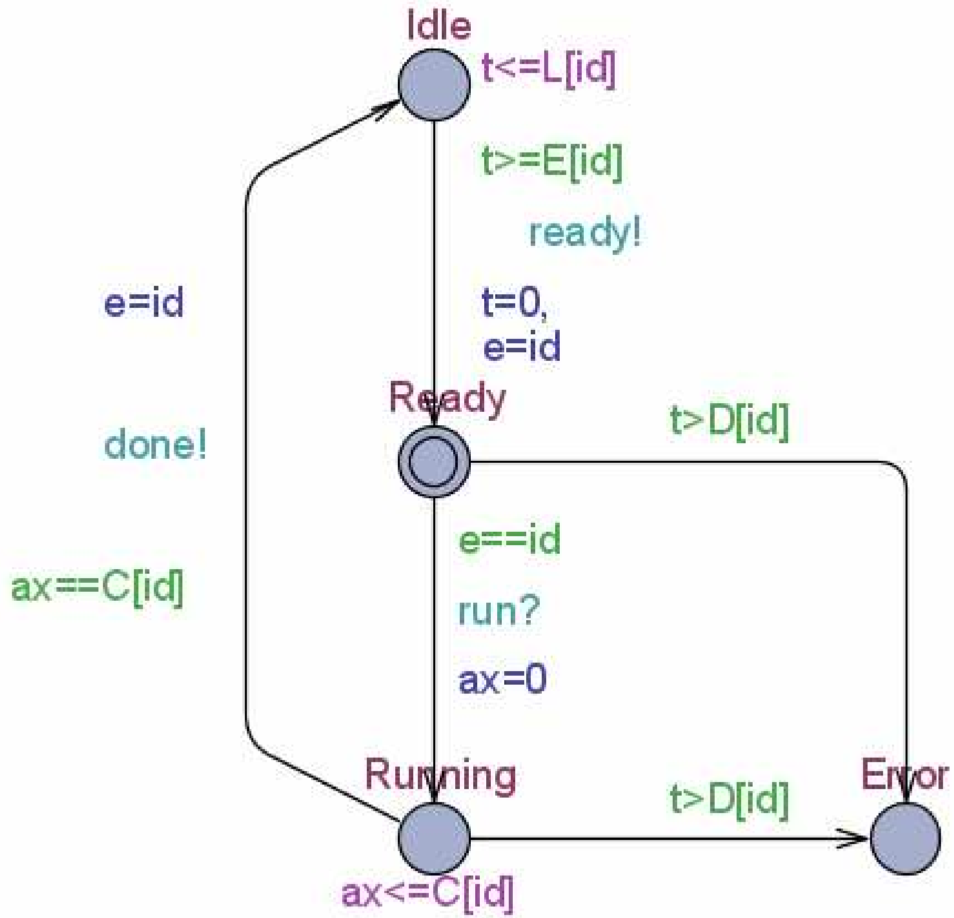


T_2 is running

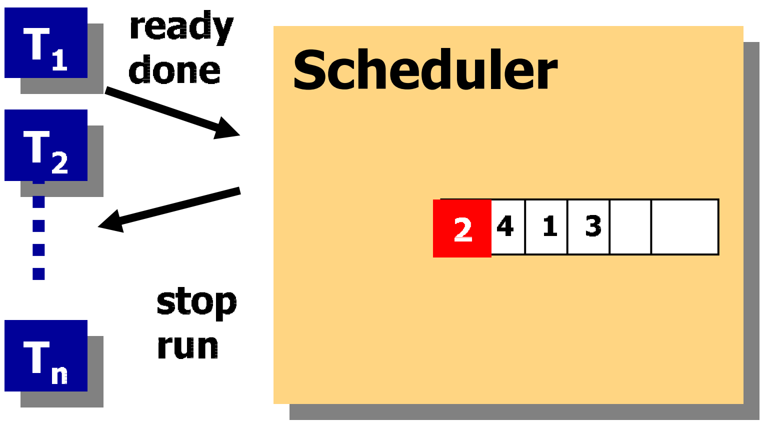
{ T_4 , T_1 , T_3 } ready
ordered according to some given priority:
(e.g. Fixed Priority, Earliest Deadline,..)

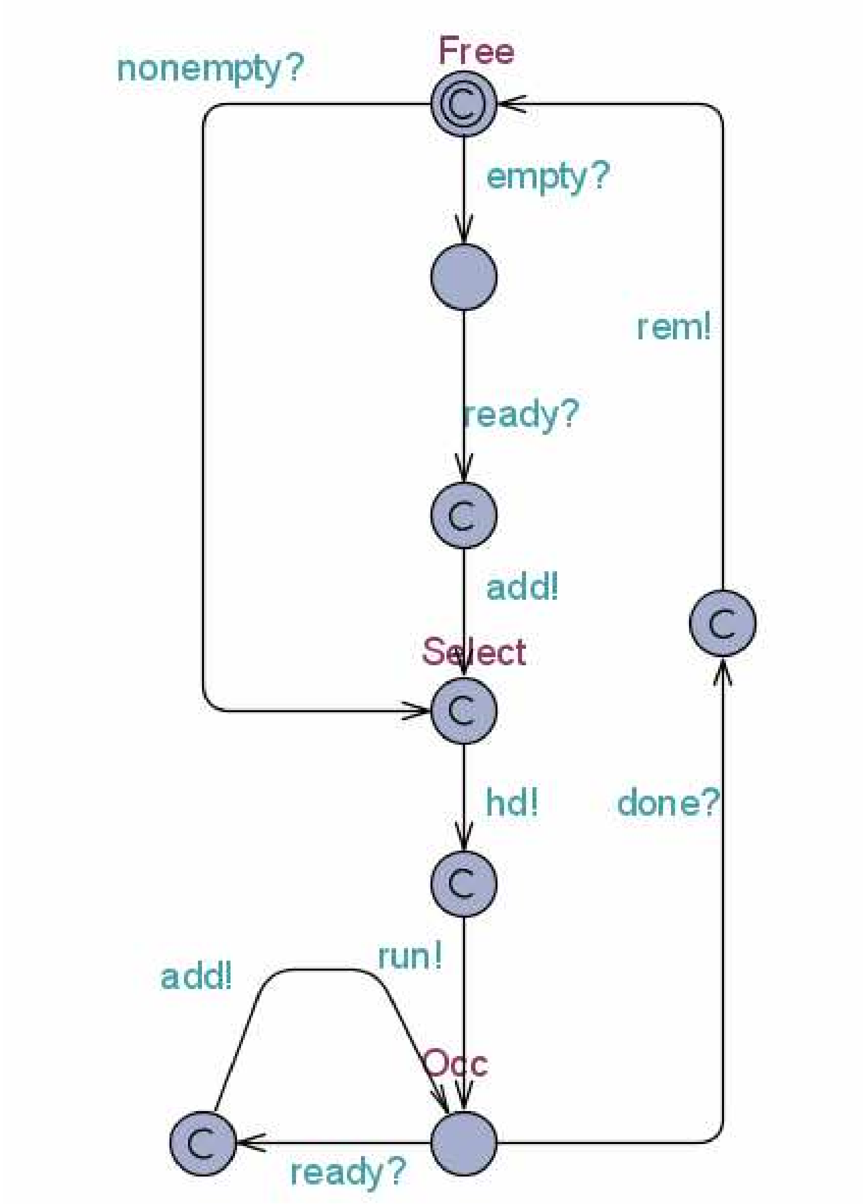
Modeling Task



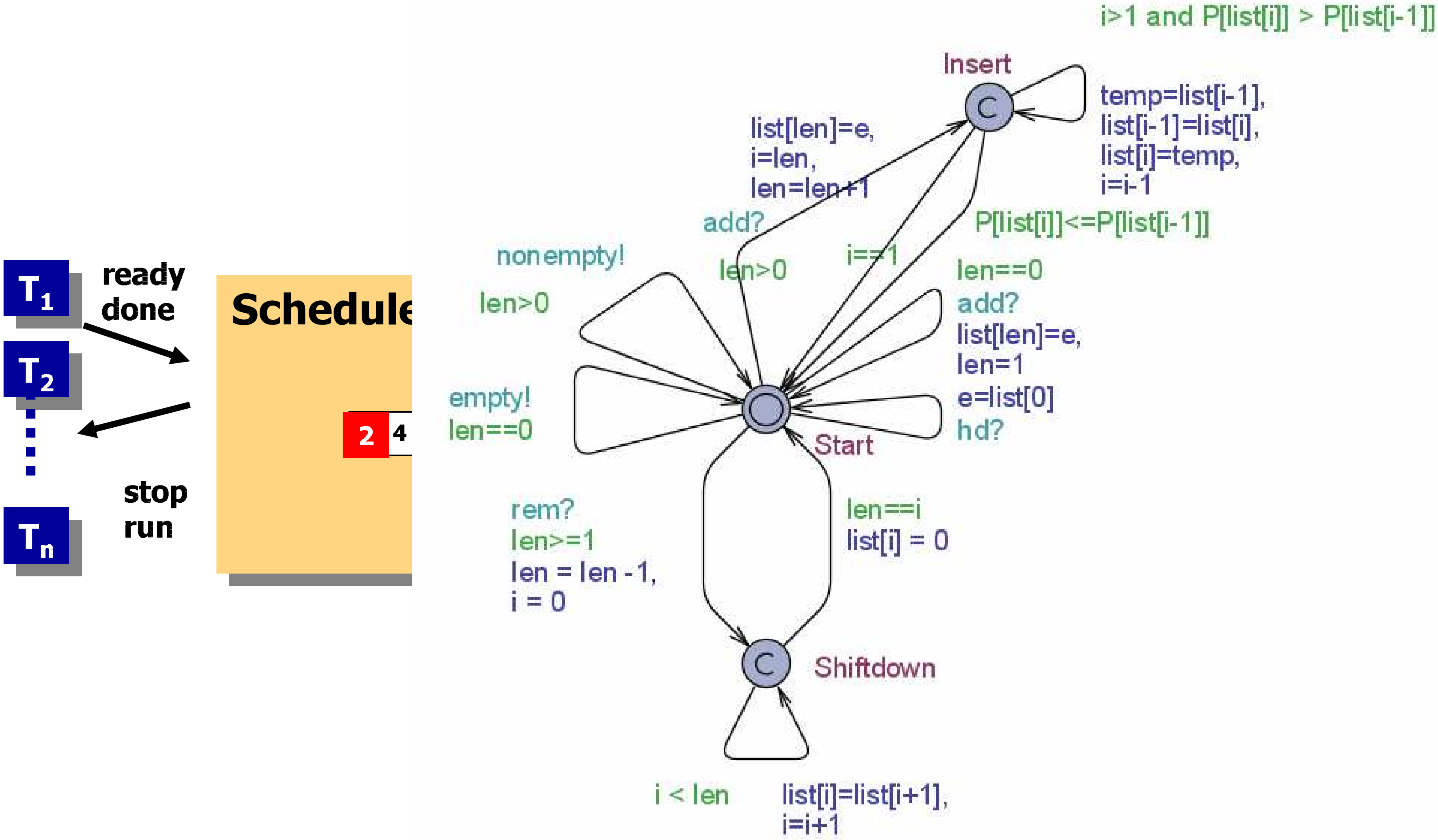


Modeling Scheduler



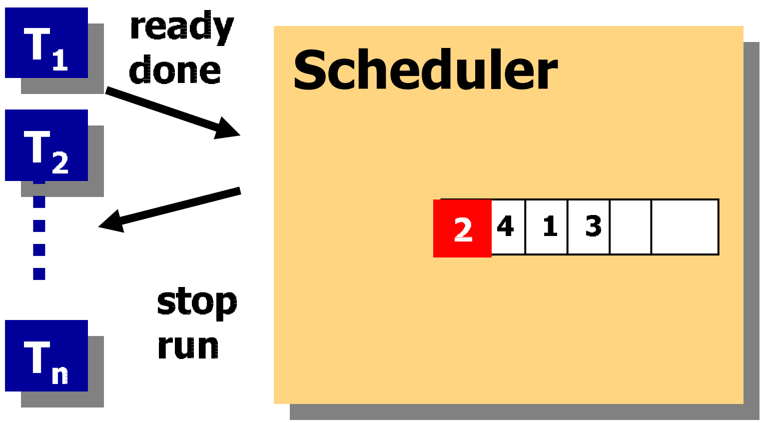


Modeling Queue



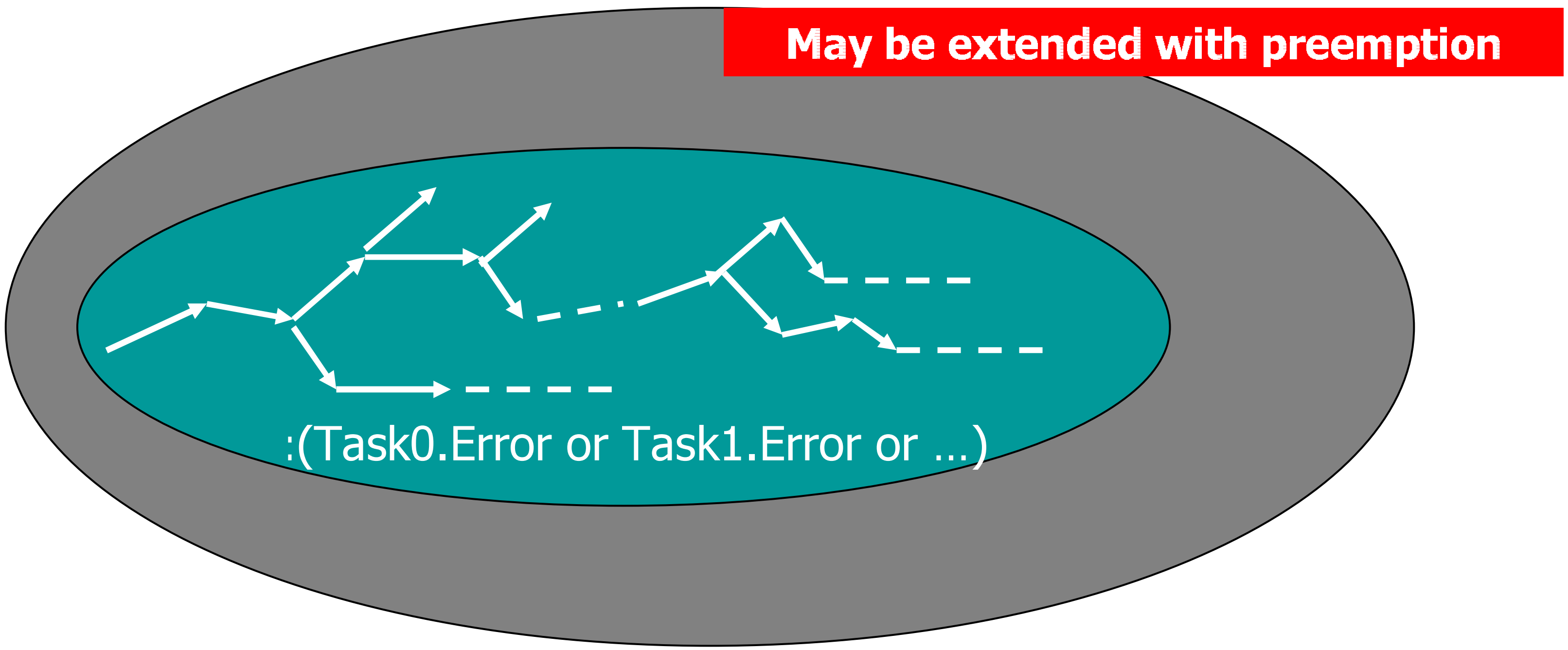
Modeling Queue

In UPPAAL 4.0
User Defined Function




```
void add(id_t element)
{
int i=len;
int temp=0;
if(len==0)
{       list[len]=e;
len=1;
}
   else
{       list[len]=e;
        i=len;
        len=len+1;
        while (i>1 && P[list[i]]> P[list[i-1]])
        {       temp=list[i-1];
                list[i-1]=list[i];
                list[i]=temp;
                i=i-1;
        }
}
}
```

Schedulability = Safety Property



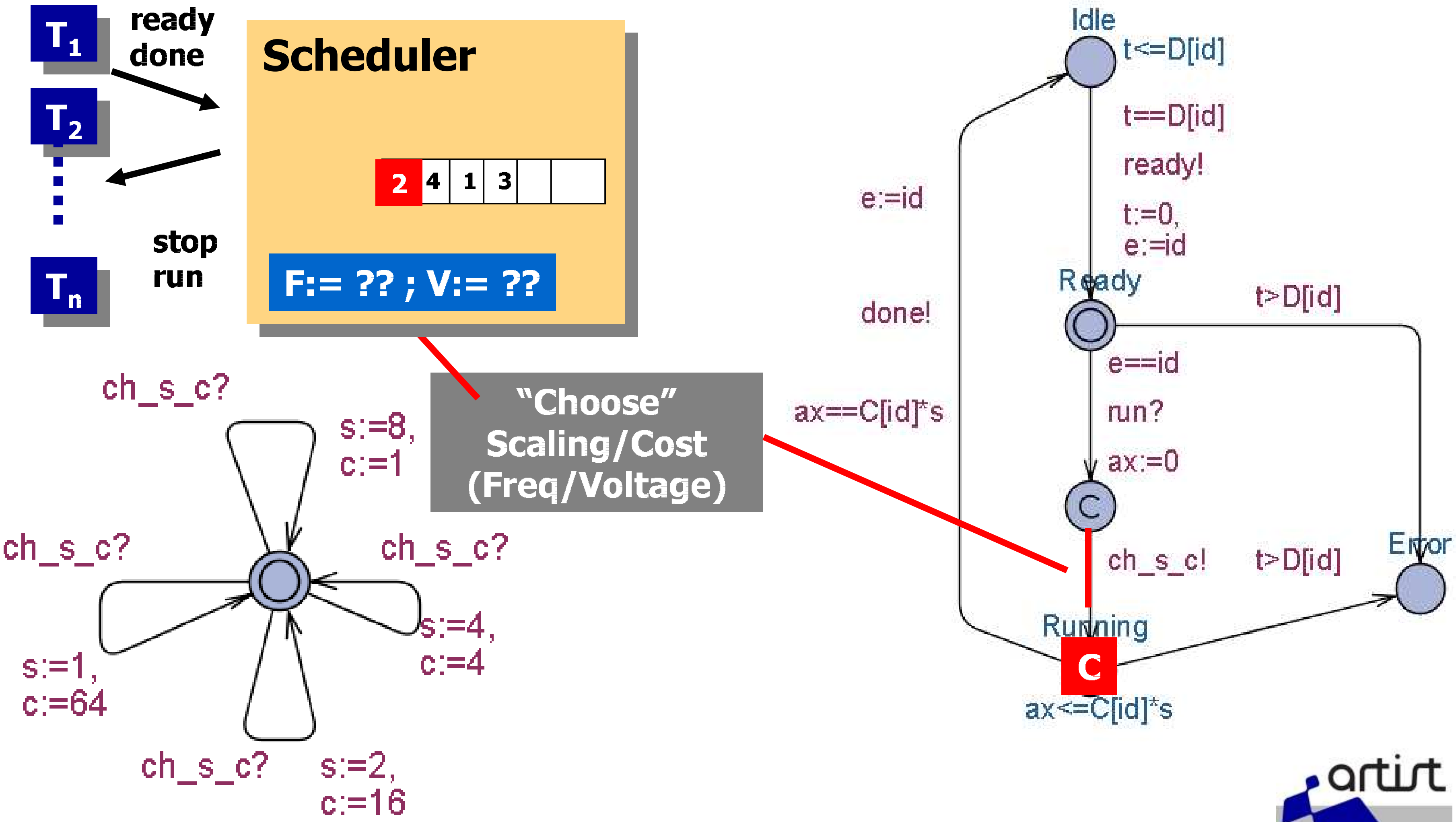
A□ :(Task0.Error or Task1.Error or ...)

CISS
CENTER FOR INDLEJREDE SOFTWARE SYSTEMER

artist

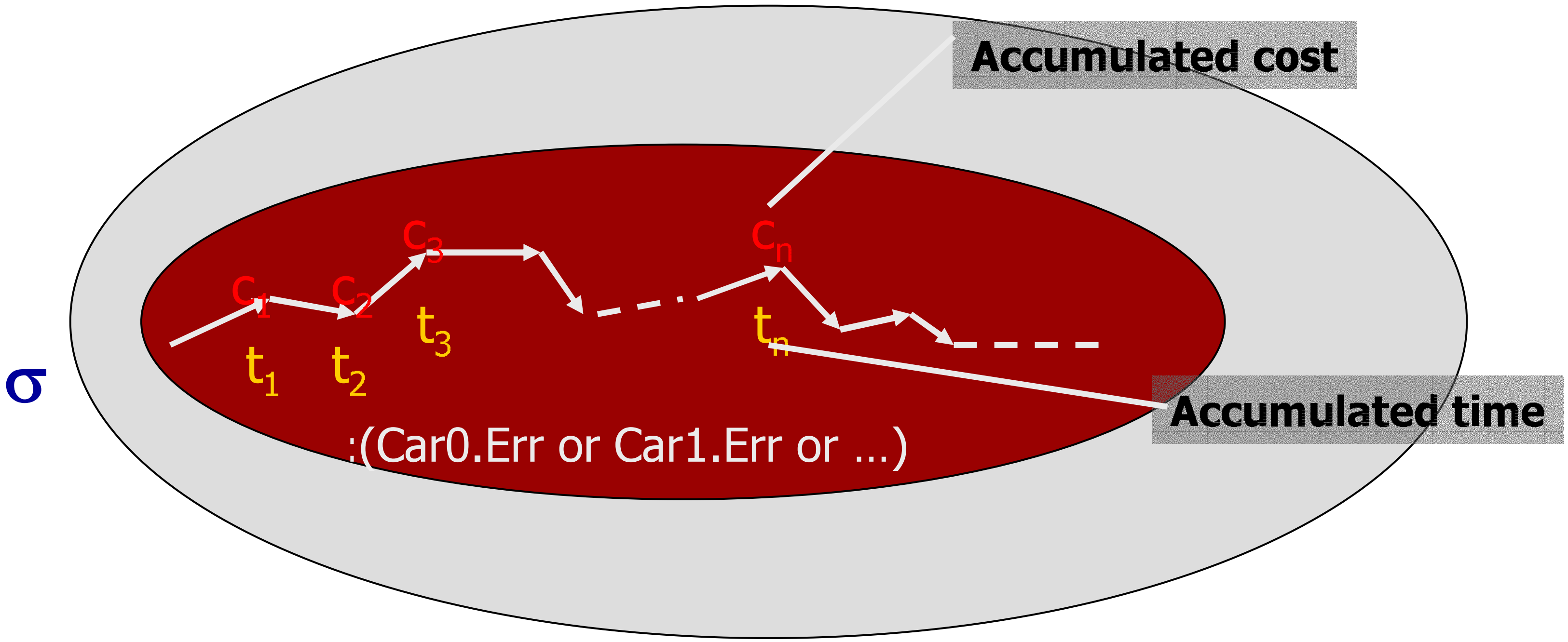
Energy Optimal Scheduling

Using Priced Timed Automata



CENTER FOR INDLEJREDE SOFTWARE SYSTEMER

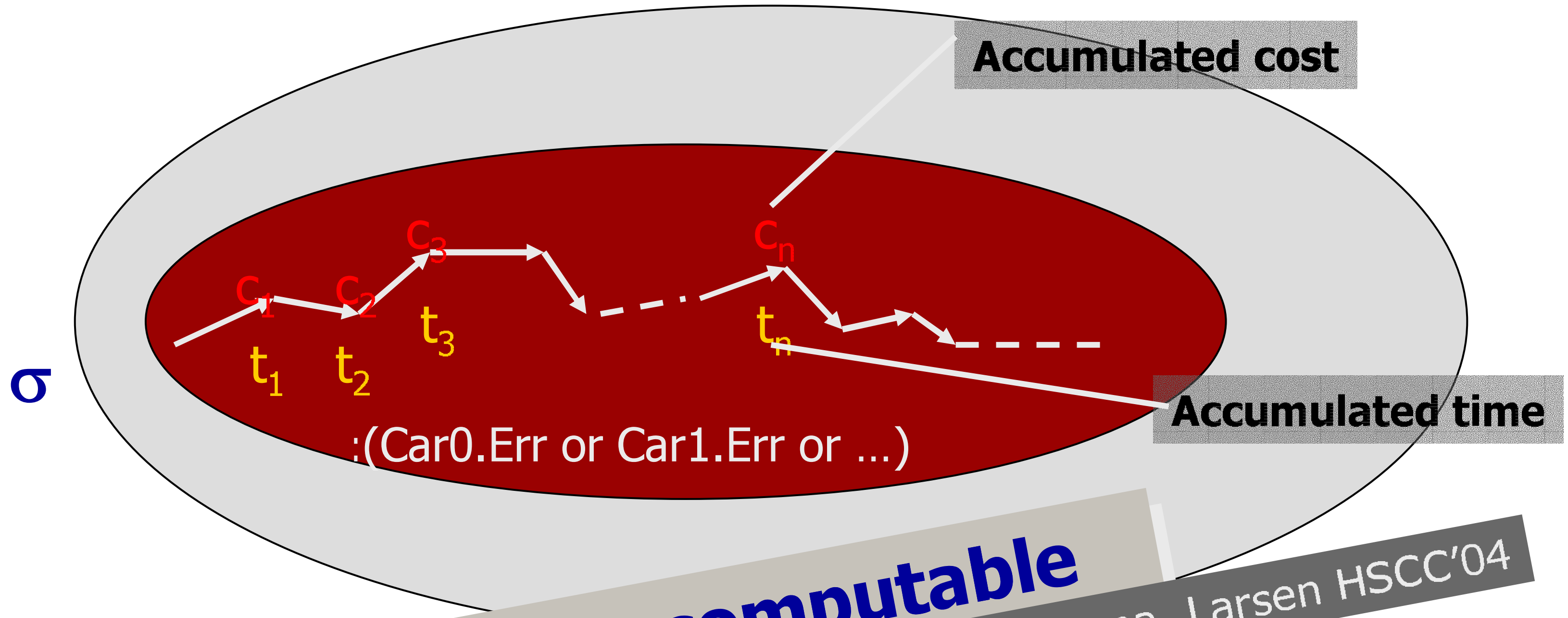
Cost *Optimal* Scheduling =
Optimal Infinite Path



Value of path σ: $val(\sigma) = \lim_{n\to\infty} c_n/t_n$

Optimal Schedule σ^*: $val(\sigma^*) = \inf_{\sigma} val(\sigma)$

Cost Optimal Scheduling = *Optimal Infinite Path*



THEOREM: σ* is computable

Bouyer, Brinksma, Larsen HSCC'04

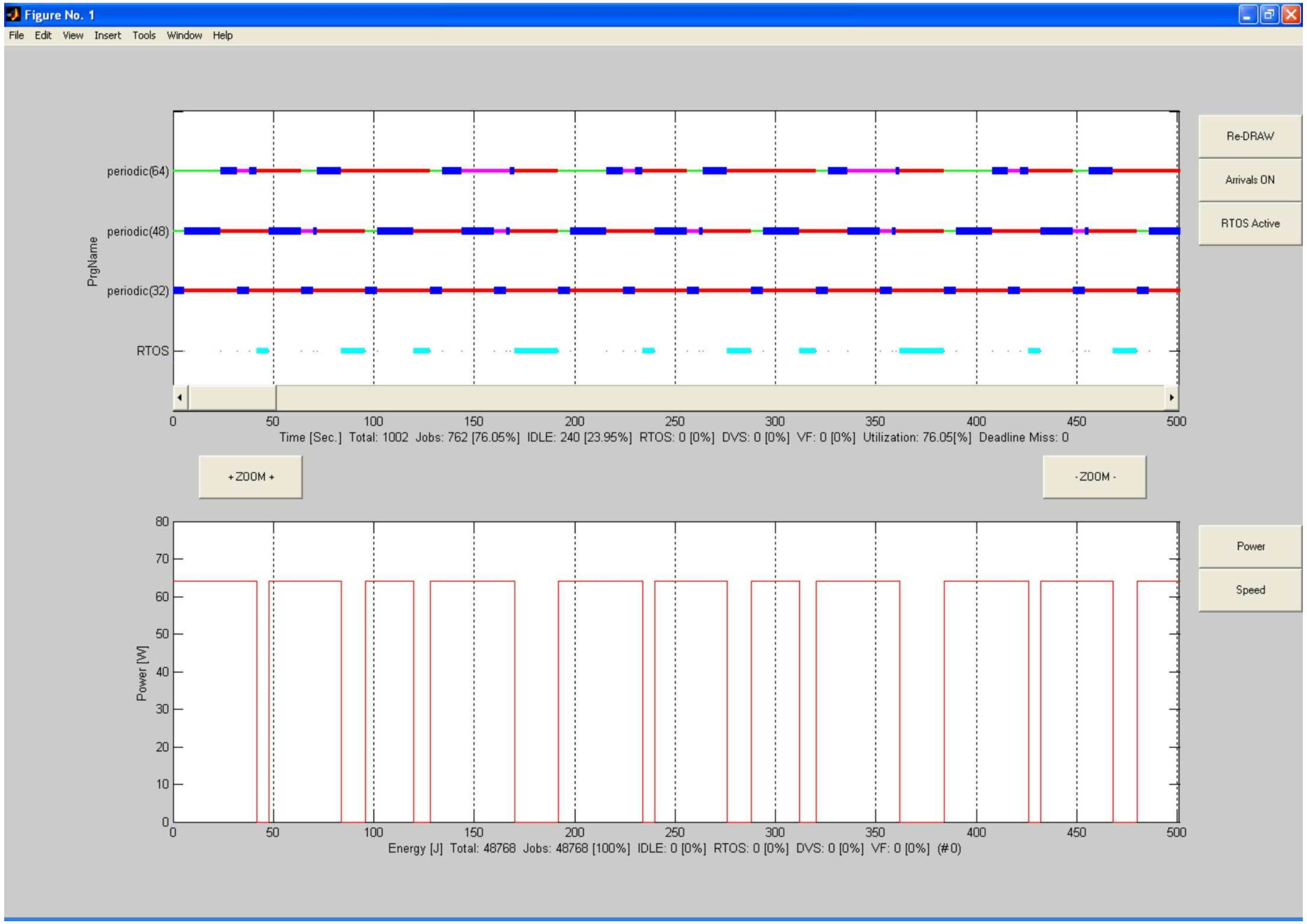
... path σ: val(σ) = $\lim_{n!1} c_n/t_n$

Optimal Schedule σ*: val(σ*) = $\inf_\sigma$ val(σ)

CISS CENTER FOR INDLEJREDE SOFTWARE SYSTEMER

artist

Preliminary Results



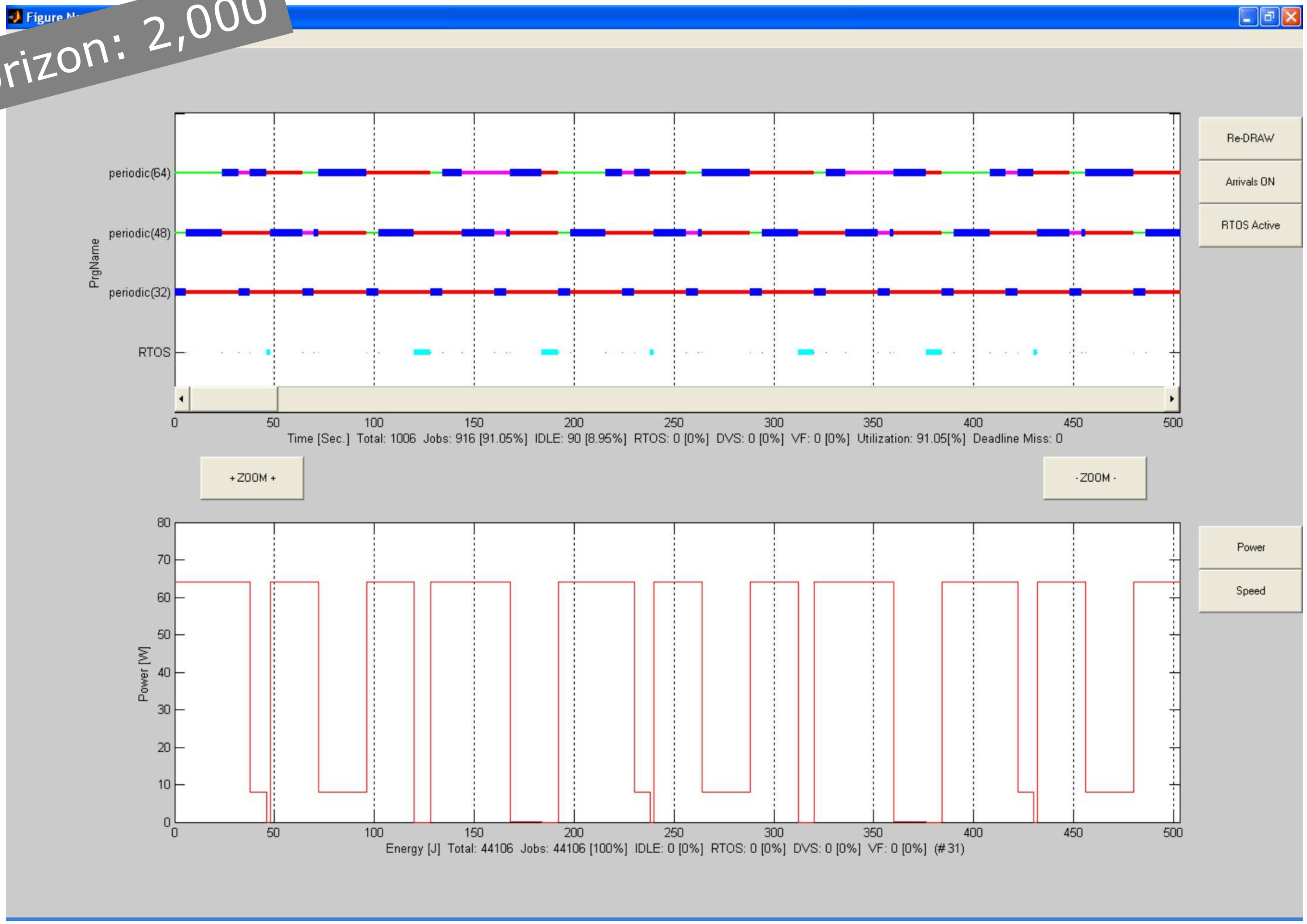
EDF w preemption no DVS: avr.: 48

P_1=D_1=32 C_1=6
P_2=D_2=48 C_2=18
P_3=D_3=64 C_3=12

artist

Preliminary Results

Cost horizon: 2,000



EDF w preemption w DVS: avr.: 43.37

$P_1=D_1=32$ $C_1=6$
$P_2=D_2=48$ $C_2=18$
$P_3=D_3=64$ $C_3=12$