# [SI] 2016

<https://twitter.com/pakalupapito/status/688682948737019904>

***Subiect 1***

**1. Avem 3 procese periodice P1=(1, 4, 4), P2 = (1, 3, 3) si P3 = (2, 6, 6), unde fiecare proces este determinat de (c, p, d) (c = timpul de executie, p = perioada, d = deadline). Care este utilizarea procesorului?**

Utilizarea e U = ¼ + ⅓ + 2/6 = 0.91

U=c/p - scrie in **curs6,slide37**

**2. Pentru exemplul anterior, este posibil ca un algoritm RMS sa produca o planificare fezabila? Daca da, dati un exemplu de o astfel de planificare.**

<https://www.ida.liu.se/~unmdu17/RTS_CUGS_files/Lecture2.pdf>

**slide 13 / 18 pdf-ul din link**

P1=(1, 4, 4), P2 = (1, 3, 3) si P3 = (2, 6, 6)

U = ¼ + ⅓ + 2/6 = 0.91 <1

Dar trebuie sa indeplineasca conditia U <= n\*(2^1/n − 1) = 3\*((radical de ordin 3 din 2) -1) = 0.78.

Asta inseamna ca este posibil ca planificarea produsa sa nu fie fezabila, dar trebuie incercat.

P1: \_ x \_ \_ | x \_ \_ \_ | x \_ \_ \_ |

P2: x \_ \_ | x \_ \_ | x \_ \_ | x \_ \_ |

P3: \_ \_ x \_ \_ x | \_ x \_ \_ x \_ |e

**3. Acelasi lucru cu EDF.**

<https://www.ida.liu.se/~unmdu17/RTS_CUGS_files/Lecture2.pdf> **slide 26 / 31 pdf-ul din link**

P1=(1, 4, 4), P2 = (1, 3, 3) si P3 = (2, 6, 6)

U = ¼ + ⅓ + 2/6 = 0.91 <1

Explicatii: La fiecare cuanta de timp se alege task-ul care are deadline-ul cel mai aproape. La egalitate, eu am vazut asa in exemplu: daca cel pe care il rulezi mai are de rulat si e unul care are deadline exact odata cu el, il termini pe cel in lucru, apoi il pui pe celalt sa ruleze. La noi e un caz la final in care se termina P3 si ai de ales intre P1 si P2, care au acelasi deadline. Eu am ales 1, apoi 2. Nu cred ca e vreo chestie impusa, mi s-a parut mai logic asa. Vreau sa imi spuneti daca si voi credeti ca e bine tot ce am facut. Vine imediat si RMS. Acolo prioritatile sunt dupa perioada cea mai scurta, cel mai prioritar e cel cu perioada cea mai scurta, ca la SO, ca se termina mai repede si ii lasa si pe altii. :D thanks :) oricand :) :) //cine a scris explicatiile? :D

P1: \_ x \_ \_ | \_ x \_ \_ | \_ x \_ \_ |

P2: x \_ \_ | \_ x \_ | x \_ \_ | \_ x \_ |

P3: \_ \_ x x \_ \_ | \_ x x \_ \_ \_ |

**4. Evaluati urmatoarea expresie in cod ce ruleaza pe o zero-address machine: G = (A - B/C) \* (D - E/F) si estimati numarul total de accese la memorie.**

**Cursul 2 -> pagina 10**

push f 2

push e 2

div 1

push d 2

sub 1

push c 2

push b 2

div 1

push a 2

sub 1

mult 1

pop g 2

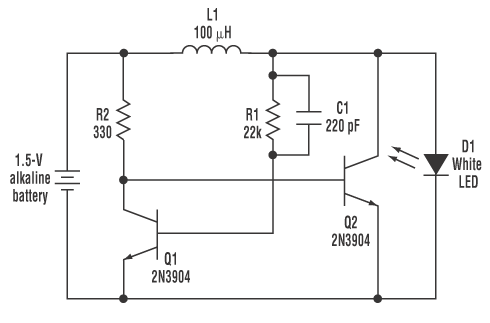
**19 accese la memorie**

**5. Proiectati si explicati functionarea unui convertor DC-DC.**

Convertoarele DC/DC sunt surse de alimentare care convertesc puterea electrică având o tensiune DC nestabilizată la intrare, într-o tensiune DC stabilizată la ieşire, cu nivel mai mic (step-down), mai mare (step-up), egal (stable level) sau cu polaritate inversată (invert) faţă de tensiunea de intrare.

si proiectarea de unde o luam?- nu am gasit

--sunt pe net diverse modele. Probabil o doreste pe aia basic cu tranzistoare & co.

**Nu e sigur ca asta e ce trebuie...**

**6. Care componenta a unui controller PID este responsabila pentru variatia timpului de atingere a referintei?**

componenta diferentiala (D) ??? da,si e singura care e in raport cu timpul asa ca ea e responsabila pentru variatia timpului

**7. Dati un exemplu de sistem de control in bucla deschisa. Cum se comporta aceste controllere intr-un sistem de timp real?**

**curs7 - slide 8** ex: Cruise-control: frecare(t), unghi\_plan(t) - tot acolo explica cum se comporta

**8. Iesirea unui tahometru este 0V pentru 0rpm si 5V pentru 1000rpm. Care este functia de transfer a tahometrului?**

probabil H = variatia iesirii / variatia intrarii = (5-0) / (1000 - 0)

**9. Se da urmatoarea succesiune de procente de utilizare a unui procesor: 0->0,2->0,9->0,1->0,5->0,6->0,7->0,8->0,6->0,4->0,3->0,1. Care este valoarea estimata a utilizarii procesorului pentru urmatoarea cuanta de timp folosind algoritmii LONG\_SHORT si CYCLE?**

LONG\_SHORT:

[0+0.2+0.9+0.1+0.5+0.6+0.7+0.8+0.6+4\*(0.4+0.3+0.1) ]/[9+ 4\*3]=7.6/21=0.36

=> fclk = 0,36 \* f max

**Aged Average**

average = ⅓ \* 0,1 + 2/9 \*0.3 + 4/27 \* 0,4 + 8/81 \*0.6 +.....

=> fuclk = average \* fmax

CYCLE:

**(la fiecare val\_i, cu i > 1, taiem cate doua valori (cele mai vechi, cele din stanga) si pe restul le impartim in doua si aplicam formula**:

val1 = (|0-0.7| + |0.2-0.8| + |0.9-0.6| + |0.1-0.4| + |0.5-0.3| + |0.6-0.1|) / 6 = 0.43 > 0.2

val2 = (|0.9 - 0.8| + |0.1 - 0.6| + |0.5 - 0.4| + |0.6 - 0.3| + |0.7 - 0.1|) / 5 = 0.32 > 0.2

val3 = (|0.5 - 0.6| + |0.6 - 0.4| + |0.7 - 0.3| + |0.8 - 0.1|) / 4 = 0.35 > 0.2

val4 = (|0.7 - 0.4| + |0.8 - 0.3| + |0.6 - 0.1|) / 3 = 0.43 > 0.2

val5 = (|0.6 - 0.3| + |0.4 - 0.1|) / 2 = 0.3 > 0.2

val 6 = |0.3 - 0.1| = **0.2** = 0.2 hurraaay :)) => ciclul era format din ultimele doua valori, 0.3 si 0.1 (ne imaginam o bara intre ele :)) ) si luam valoarea de inceput a celei de-a doua parti, adica 0.1, la care adunam **0.2 (acel 0.2 cu verde, putea sa fie oricat, cat timp era <= 0.2)** => valoarea estimata este 0.1 + 0.2 = 0.3 (nu este media primelor 2 valori din fiecare ciclu + eroarea? )

Daca si alte val\_i erau mai mici decat 0.2, il alegeam pe cel mai mic si aplicam aceeasi idee, il adunam la prima valoare din jumatatea a doua a ciclului respectiv si aia era predictia.

0.2 e constantă de algoritm, se compara cu ea mereu. Eu am înțeles combinând link-ul de sub ce scriu cu explicația- din doc-ul de la Anca. multumesc

<https://www.researchgate.net/publication/220926731_Comparing_Algorithm_for_Dynamic_Speed-Setting_of_a_Low-Power_CPU>

pe la pag 6-7 e exemplul din curs cu cycle ( si nu numai ) insotit de ceva explicatii

**10. Ce este un barrel shifter si la ce poate fi folosit? Care este avantajul unei structuri barrel shifter+UAL fata de o structura shifter+UAL?**

* Un barrel shifter este o unitate funcțională care permite shiftări și rotații (5 tipuri) pentru numere pe un anumit număr de biți într-un singur ciclu de ceas.
* La procesoarele ARM se folosește împreună cu UAL, prelucrând cel de-al doilea operand al UAL.12655956_886138028165557_1383078353_o.jpg
* Folosit adesea pentru operații în virgulă mobilă.
* Avantaj față de UAL + shifter: shiftează cu n biți într-un singur ciclu de ceas, spre deosebire de shifter, care are nevoie de n cicli de ceas.

***FI-MI-AR SCARBA!!! ...si mie..cui nu***

11. Problema cu lamaia12655956_886138028165557_1383078353_o.jpg

7 lei kg lamai, pp 10 lamai pe kg => 700000(156k euro) lei pt 100 W12655956_886138028165557_1383078353_o.jpg12655956_886138028165557_1383078353_o.jpg12655956_886138028165557_1383078353_o.jpg12655956_886138028165557_1383078353_o.jpg

***Subiect 2***

**1. Zero adrress machine A=B\*C-D+E**

push D 2

push C 2

push B 2

mult 1

sub 1

push E 2

add 1

pop A 2

total: 13

**2. regula 80-20**

80% din timp sunt folosite 20% din instrucțiunile existente în setul de instrucțiuni al unui procesor. :) -curs 2

**3. baterie 1Ah… curs 3, slide 46 (formula Peukert)**

**<https://www.youtube.com/watch?v=uWb2ZZzDBFw> ->totul despre baterii**

--O baterie care functioneaza la 1Ah se descarca cu 4C pe o rezistenta de 2

In cat timp se descarca complet ?

--o baterie de 1Ah se consuma intr-o ora cu 1C (unde apare proprietatea asta? si de ce nu se foloseste deloc rezistenta in rezolvarea problemei?) cred ca am gasit unde se foloseste rezistenta (vedeti in imagine)

cu 4C se va consuma in 15 min (60min / 4) <- da, asa e

Am sters poza l-am intrebat pe vladimir si 15 min e :))))

**4. breakeven point**

**Cursul 5 -> pagina 24**

=timpul minim de idle pentru care e eficient sa se faca shutdown.

**5. mostenire si inversiune prio in multitasking**

<https://www.ida.liu.se/~unmdu17/RTS_CUGS_files/Lecture3.pdf> (maybe)

**Curs 9 slide 53**

**6. 3 metode de colectare a energiei + cea mai eficienta dpdv a energiei captate**

1.Energy Harvesting Fotovoltaic(cea mai eficienta)

-Foloseste celule de siliciu pentru a transforma energia luminoasa in energie electrica

-are un randament de 10-13%

2 .Energy Harvesting Piezoelectric

-grinda elastica de material piezolectric are atatsata la capatul mobil o greutate,si prin oscilatia acelei greutati se produce energie.

3.Energy Harvesting Termoelectric

-se foloseste efectul Seebeck,care genereaza energie electrica din diferenta de temperatura.

-trebuie folosit intr-un mediu unde exista diferente de temperatura > 30 C

**7. proprietati kernel RTOS**

**Cursul 9 -> pagina 12**

–Un sistem embedded este proiectat pentru un singur scop asa ca majoritatea functionalitatilor unui SO comercial sunt redundante (consola, interfata grafica, suport tastatura, mouse etc.).

–RTOS permite controlul ferm asupra resurselor sistemului

•Nu exista procese de background inutile

•Numar maxim de task-uri care pot rula pe sistem

–RTOS permite controlul temporizarii proceselor

•Manipularea prioritatii task-urilor

•Optiuni de setare a mecanismului de planificare

**8. LUT pentru (a&b)|c**

* tabela de adevar pentru expresia asta, ca la PL <http://yosefk.com/img/n/lut3.png>

**9. avantaje si dezavantaje procesor SW**

**10. care kernel e mai bun pentru embedded + motivatie**

Kernelul monolitic este mai bun pentru sistemele embedded deoarece

- dispune de o multitudine de drivere,dintre care sunt incarcate doar cele care iti trebuie

-este foarte versatil,adica e compatibil cu o paleta larga de deviceuri

(laboratorul cu Kernel)

***Subiect 3***

**1. Comparatie RISC - CISC**

|  |  |
| --- | --- |
| **RISC** | **CISC** |
| pune accentul pe software | … hardware |
| instructiuni reduse intr-un singur ciclu de ceas | fix pe dos |
| operatiile de load si store sunt implementate ca instructiuni separate | se regasesc in alte instructiuni complexe (ex: add) |
| cod cu multe linii | putine linii |
| complexitate redusa - nr mic de tranzistori pe chip - lasa loc de periferice | complexitate mare - nr mare de tranzistori |
| necesita un spatiu marit de memorie pentru cod si date | nu necesita memorie multa |

**2. Prezentare PID**

* cel mai folosit controller din industrie datorita stabilitatii, timpului mic de raspuns si aproximarii erorii
* format din 3 parti:
  + (P) - controller proportional
  + (PD) - controller proportional diferential
    - (D) - timp mic de raspuns, sensibil la zgomot
  + (PI) - controller proportional integral
    - (I) - tracking si rejectia perturbatiilor, poate produce oscilatii
* + formula :(
* + desen :(

**3. Prezentare FPGA**

• Circuitele FPGA sunt folosite pentru generarea

circuitelor logice complexe.

• Un chip conține un număr foarte mare (zeci de mii) de

blocuri logice configurabile.

– Programele de tip CAD mapează circuitele de nivel înalt peste

matricea de blocuri de bază prin configurarea generatoarelor

de funcții, interconexiunilor și altor elemente configurabile

• Blocurile logice sunt rutate folosind interconexiuni

programabile

– Segmentele sunt conectate la blocurile logice și la alte

segmente învecinate prin switch-uri configurabile

– Programele CAD determină configurația optimă pentru toate

switch-urile folosite.

# **4. Prezentare TinyOS**

* imagine binara executata pe ATMEGA
* arhitectura event-driven
* nu are kernel, nu are managment al proceselor, al memoriei sau memorie virtuala
* planificare simpla de tip FIFO
* scheduling
  + pe 2 nivele - evenimente si taskuri
  + stiva partajata intre evenimente si apeluri de functii
* taskuri
  + nu pot fi preemptate de alte taskuri dar pot fi preemptate de alte evenimente
  + apeleaza comenzi
  + detecteaza evenimentele
* evenimente
  + sunt implementate de intreruperi hardware ale procesorului

**5. 2 strategii hardware de reducere a consumului de energie in sistemele incorporate**

-Reducerea tensiunii de alimentare (V)

-Reducerea activității (A)

-Reducerea curenților reziduali Curs 4 slide 28 ?

**6. Principalele functii ale unui kernel RTOS + scurta prezentare**

**Cursul 9 -> pagina 16**

–Task Scheduler : Determina care task va rula in cuanta de timp urmatoare pentru un sistem multitasking.

–Task Dispatcher: Produce informatia necesara in contextul pornirii unui task

–Intertask Communication: Implementeaza un mecanism de comunicatie intre doua procese

Curs 1, slide 49

**8. Principiul dupa care functioneaza recoltarea de energie capacitiva**

**Cred ca iti faci un condensator care isi schimbă A si d in funcție de vibratii, astfel pt Q constant V creste**

**9. Definiti conceptul Smart Dust**

Nodurile dintr-o retea WSN (Wireless sensor network) vor ajunge la dimensiuni microscopice(graunte de praf).

ce am gasit eu cautand pe google:  
MEMS = micro-electo-mechanical, tehnologia dispozitivelor f mici de ordinul micrometrilor

Smartdust= un ansambul de MEMS pura, vibratii) precum senzori, roboti, etc care pot detecta diversi parametri (lumina[, temperatura) si care actioneaza de obicei intr-o retea wireless si sunt distribuiti pe diverse zone pt a rezolva sarcini.](https://www.fourmilab.ch/hackdiet/www/subsection1_2_3_0_5.html)

[**10. Diferente intre controllerul cu 2 praguri si c**](https://www.fourmilab.ch/hackdiet/www/subsection1_2_3_0_5.html)**el proportional**

[intre controllerul P (1 prag) si controllere de genul PI sau PD (2 praguri?) ?](https://www.fourmilab.ch/hackdiet/www/subsection1_2_3_0_5.html)

Aici se refera cumva la o diferenta(controller cu 2 praguri)

[Se refera la cel proportional vs Bang bang](https://www.fourmilab.ch/hackdiet/www/subsection1_2_3_0_5.html)

**Subiect 4**

**1. ubiquitous computing**

**Cursul 1 -> pagina 39**

Model de interacțiune om-masină în care procesarea informațiilor este adânc integrată în obiectele și activitățile zilnice.

Principii :

•The purpose of a computer is to help you do something else.

•The best computer is a quiet, invisible servant.

•The more you can do by intuition the smarter you are; the computer should extend your unconscious.

•Technology should create calm.

**2. clasificare WSN**

**Cursul 12 -> pagina 11**

• Environmental Sensor Networks (ESN):

• număr mare de noduri

• constrângeri severe de consum de energie

• Community Sensor Networks (CSN):

• număr mediu de noduri

• atașate la o infrastructură deja existentă

• Body Sensor Networks (BSN):

• număr mic de noduri

• atașate, sau în imediata vecinătate a unei persoane

**3. pe legea consumului CMOS sa zicem 3 strategii de reducere a consumului**

* reducerea frecventei de ceas
  + Scade puterea medie dar mărește timpul de execuție -> consumul de energie rămâne constant
* reducerea tensiunii de alimentare
* reducerea activitatii
  + Dezactivarea unor blocuri funcționale atunci când nu sunt active

Cursul 3, slide 61

**4. modelarea consumului la transceiver**

**5. kernel monolitic**

**Cursul 9 -> pagina 14**

Sper ca poza aia sa fie de ajuns.

**6. o problema cu bateria 1Ah 4C**

o baterie de 1Ah se consuma intr-o ora cu 1C

cu 4C se va consuma in 15 min (60min / 4)

**7. o relatie A+C\*... in zero address machine**

**8.super condensatorul (si schema)**

• Tehnologie emergentă

• Stochează sarcina electrostatic, sub formă de ioni

• Capacităși foarte mari (1F-1000F)

• Timpi reduși de încărcare și descărcare

• Nu se degradează cu trecerea timpului

• Preț (mai) ridicat

Cred ca schema e la cursul 11 -> pagina 46

**10. RMS vs EDF ?**

**In cadrul RMS se alege task-ul cu prioritatea cea mai mica (din multimea de task-uri released, adica ce sunt gata de executie)**

**In cadrul EDF se ia task-ul care are cel mai curand deadline**

**Curs 9, slide 48**

**http://www.cs.virginia.edu/~zaher/classes/CS656/pillai.pdf**

***Subiect 5***

**11. 2 procesoare P1 si P2 cu frecvente +tensiuni date si Ileak = Ishort =0. Care procesor este mai eficient dpdv al energiei consumate?**

**12.dintre VLIW si RISC ce ai alege, stiind ca esti constrans de memorie si de ce?**

VLIW deoarece RISC-ul necesita multa memorie. Arhitecturile VLIW au un grad foarte mare de paralelism -> reducerea consumului

Curs 4, slide 40

**13 In ce consta DVS(Dynamic Vlotage Scaling)? Explicati pe scurt motivele pentru care este utila, avand in vedere tehnologia CMOS.Care sunt limitarile sale?**

Dynamic voltage scaling is a [power management](https://en.wikipedia.org/wiki/Power_management) technique in [computer architecture](https://en.wikipedia.org/wiki/Computer_architecture), where the voltage used in a component is increased or decreased, depending upon circumstances. Dynamic voltage scaling to increase voltage is known as overvolting; dynamic voltage scaling to decrease voltage is known as undervolting . Undervolting is done in order to [conserve power](https://en.wikipedia.org/wiki/Energy_conservation), particularly in [laptops](https://en.wikipedia.org/wiki/Laptop) and other mobile devices,[[1]](https://en.wikipedia.org/wiki/Dynamic_voltage_scaling#cite_note-1) where energy comes from a battery and thus is limited, or in rare cases, to increase reliability. Overvolting is done in order to increase computer performance.

**14 explicati cum fct sist cu reactie**

**15 comparati P cu PI**

**16 prezentati sumar alg RMS .**

**18 avantaje VLIW fata de CISC**

[http://www.icverification.com/Archit](http://www.icverification.com/Architecture/CpuArch4.php)

[ecture/CpuArch4.php](http://www.icverification.com/Architecture/CpuArch4.php)

**19 comp PID cu P**

**20 ce se reduce la clock gateway?**

Puterea de comutatie si de scurtcircuit => puterea dinamica

***Subiect 6***

**21. Cerinte non-functonale.**

Cerinte non-funcționale:

– Preț, Dimensiuni, Greutate -> Puțină memorie

– Consum baterie/caldură -> Timp/viteză de lucru

– Deadline -> Timp minim de dezvoltare garantat

Curs 3, slide 3

**22. sa enumeri cel putin 3 avantaje pt care se folosesc CMOS**

viteza de comutatie mare , putere statica tinde la 0 , consum foarte scazut ( doar in comutatie consuma ) ;; bune ca pret, dimensiune , performante.

**23. FPGA. Descriere si relatia cu SoC**stie cineva ce am putea scrie la partea cu relatia cu SOC…?

intra in componenta SoC-urilor reconfigurabile

**24.Timere Software**

curs 9, slide 29

**25. Ce este un softprocessor? Ce avantaje si dezavantaje are fata de un procesor hard?**

**anybody here??**

[**http://electronics.stackexchange.com/questions/55377/soft-core-processors-vs-hard-core-processors**](http://electronics.stackexchange.com/questions/55377/soft-core-processors-vs-hard-core-processors)

|  |  |
| --- | --- |
| 15down vote | Soft core is implemented in FPGA fabric while Hard is implemented the same as any integrated circuit while still connected to the FPGA fabric. The comparison generally boils down to:  Hard Core Processors - 100's of MHz up to 1GHz+ of speed   * Can achieve much faster processing speeds since they are optimized and not limited by fabric speed * Fixed and cannot be modified (though it can take advantage of custom logic in FPGA fabric for processing   Soft Core Processors - 250MHz and less (usually less than 200MHz)   * Can be easily modified and tuned to specific requirements, more features, custom instructions, etc. * Multiple cores can be used (at the cost of resources) * Limited by the speed of the fabric.   In recent years, there's been a large move towards hard processor primarily due to the need for faster processing that the soft cores can't provide. When you put a hard core processor with the FPGA fabric, you can typically enable processing of a lot of data, which is needed for communication infrastructure applications (lots of GBs of data going by), for example. |

**26. Elemente controller fuzzy**

curs 8, slide 37

**29. Realizati o comparatie intre arhitecturile von Neumann si Harvard din punct de vedere al utilizarii lor in sistemele embedded**

cursul 2, slide 29 si 40

**30. Comparati cu accent pe avantaje si dezavantaje, sursele de alimentare liniare si cele in comutatie**

High efficiency, low power dissipation and high power density

(small size) - advantages of switching mode supply ;; disadvantages - cost ( higher cost )

**care cum?**

**31. Explicati pe scurt functionarea unui sistem de control in bucla inchisa(reactie)**

Foloseste sensori, are feedback, sisteme multiple mici care interactioneaza, mai complex la implementare

Probleme1

**P1. o mizerie cu WNS... ai o retea pe un drum liniar din doi in doi metri si te intreaba daca e mai bine in alta configuratie (cred...) stiu ca ti se dadea gamma=0.2**

P2. AveragedAge si Cycle pentru 12 valori Facut mai sus

**P3.o mizerie cu o capacitate... un bec ia energie de la o capacitate de 10F incarcata la 3.6V... se da tensiunea minima ca becul sa functioneze 1.8V si curentul 1mA. cat timp lumineaza becul?**

((3.6 V - 1.8 V) \* 10 F) / 0.001 = 18000 secunde <- e bine.

Pentru cine se indoieste de abilitatile mele de SI-ist: <http://electronics.stackexchange.com/questions/4951/how-do-i-calculate-how-fast-a-capacitor-will-discharge>

sigur e asa?  
eu m-am gandit ca  
Tdesc = C/P;

P = Umin\*Imin = 18 \* 10^-4

T = 10^5/18 = 5500s

**P4. Fuzzy cu cutia de viteze**

***Probleme2:***

~~1. Codificare Huffman pe un sir dat.Huffman nu mai e in materie~~

2. Calculat LONG\_SHORT si AGED\_AVERAGE. Este dat un sir de 12 cuante de timp.

Facut mai sus

3. Un Fuzzy pentru climatizarea unei camere (se dau termenii "temperatura" si "umiditate" )

***Probleme3:***

un sir de cuante de timp pe un procesor si sa calculam in modelul longshort si cycle

si ca probleme -> facuta mai sus

# Probleme 2015:



Are cineva rezolvari pentru problemele de aici? Am incercat sa o rezolv pe prima, insa mi se pare ca lipsesc niste date: pentru ambele procesoare Waffer\_yield (numarul de vafe defecte) si pentru procesorul 2 parametrul alfa legat de complexitatea producerii (in curs scrie ca e de obicei 0.4). <- pai spune matrita netestata, deci nu trebuie sa calculezi die yield I guess → fara Die\_yield nu se poate determina costul matritei <- fiind netestate, die\_yield e 1, nu? → Asa am considerat si eu pana la urma :D

1 a) Cred ca e cost matrita=cost vafa/nr\_matrite

Si se calculeaza suprafata totala a vafei, se imparte la suprafata matritei si se scade un termen (pi\*diametru/sqrt(2\*arie\_matrita))   
E in curs 1, pagina 30, a 2a formula.

Ptr MIPS faci frecventa/CPI, deci 166,6 si 66,6. Primul e mai rapid de ~3 ori. <-am cautat si eu am gasit frecventa/(CPI\*10^6) <https://en.wikipedia.org/wiki/Cycles_per_instruction> <- 10^t6

2a) Ton = 1ms Toff = 1ms Pon = 100 mw Poff = 0 mw

Ttr = Ton + Toff = 2ms

Ptr = (10^-4 + 10^-4)/(2 \* 10^-3) = 0.1W

Ptr <= Pon => Tbe = Ttr = 2m