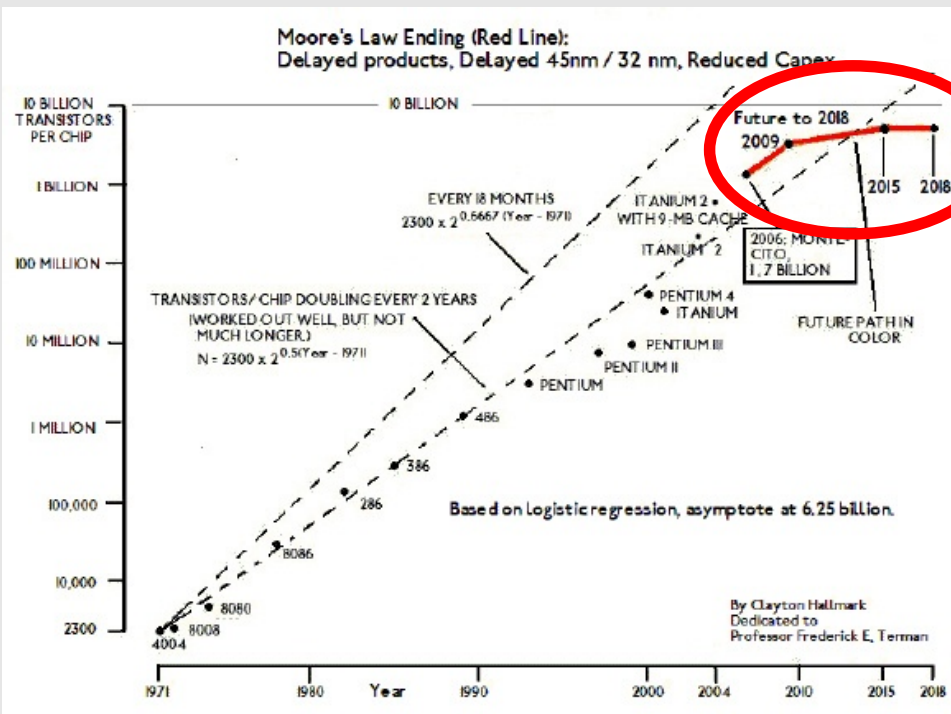


# **Numerical Simulation and Modelling for EDA: Past, Present and Future**

Jaijeet Roychowdhury

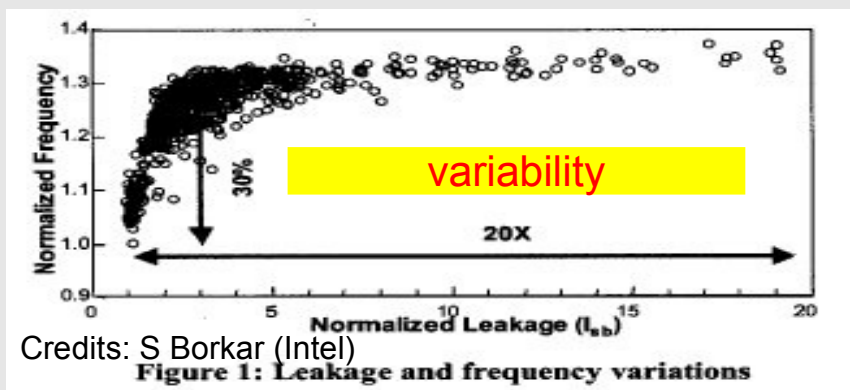
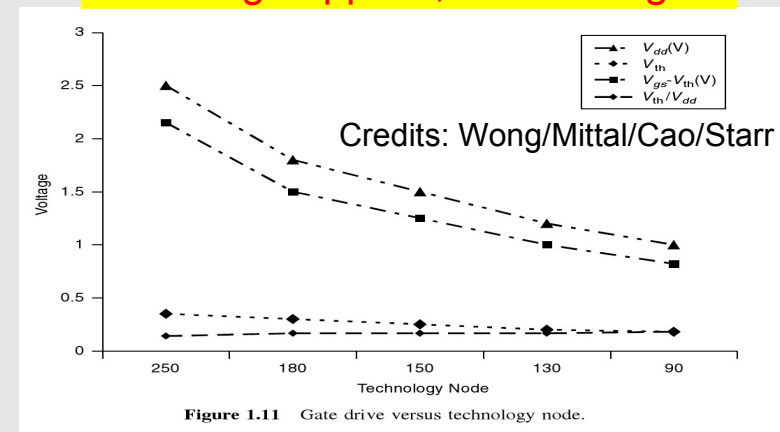
University of California, Berkeley

# Resurgence of Simulation

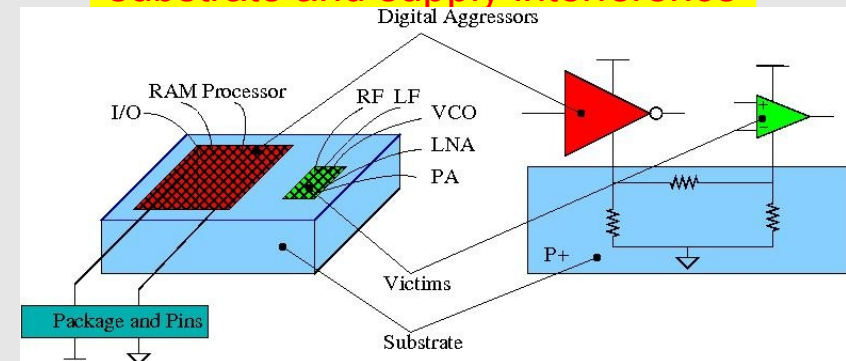


**“Analog Issues”** = breakdown of “clean” digital abstractions

shrinking supplies, noise margins



substrate and supply interference



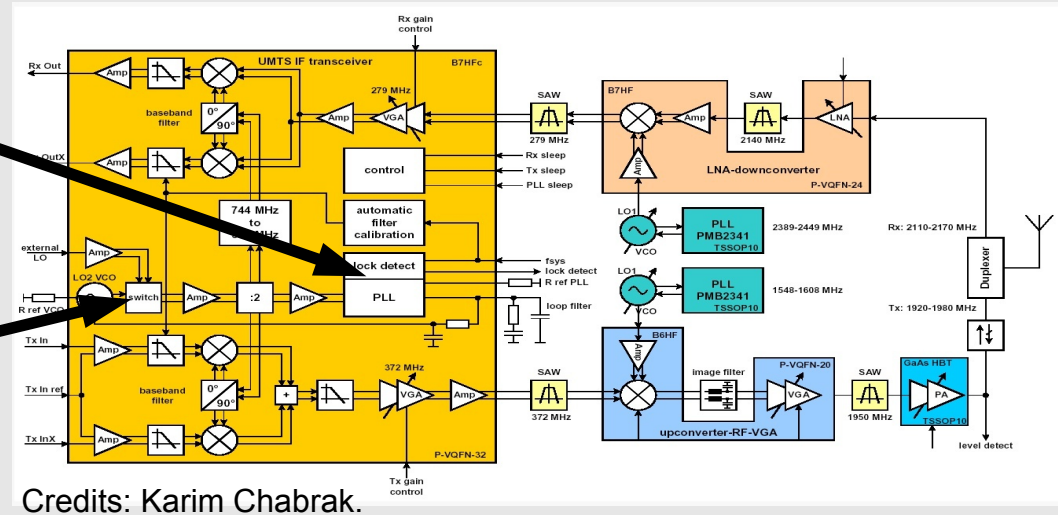
**Fact: O(months) of simulation time spent characterizing digital cell libraries**

# Numerical Simulation in EDA

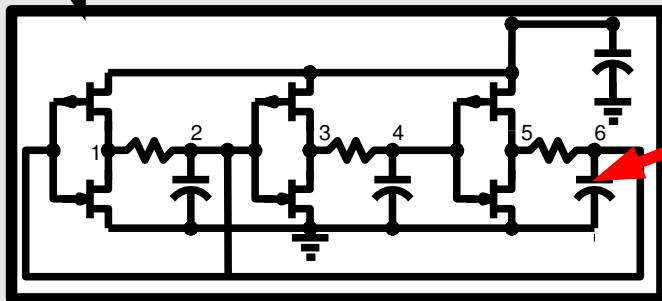
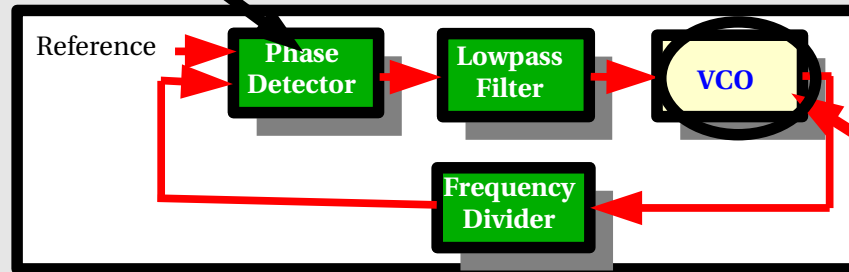
- **SPICE (analog): 70s**
  - DC, AC, transient, DC noise
- **RF: 70s-90s**
  - periodic steady state, time-varying AC, time-varying noise
- **interconnect, interference (digital): 80s-00s**
  - LTI model reduction
- **today's challenges:**
  - **variability**
  - **scale (“fast SPICE”)**
  - **proliferation of experimental devices (nano)**
  - **system-level: macromodelling**
  - **applications outside EDA**

# System Level Abstraction

High-level models



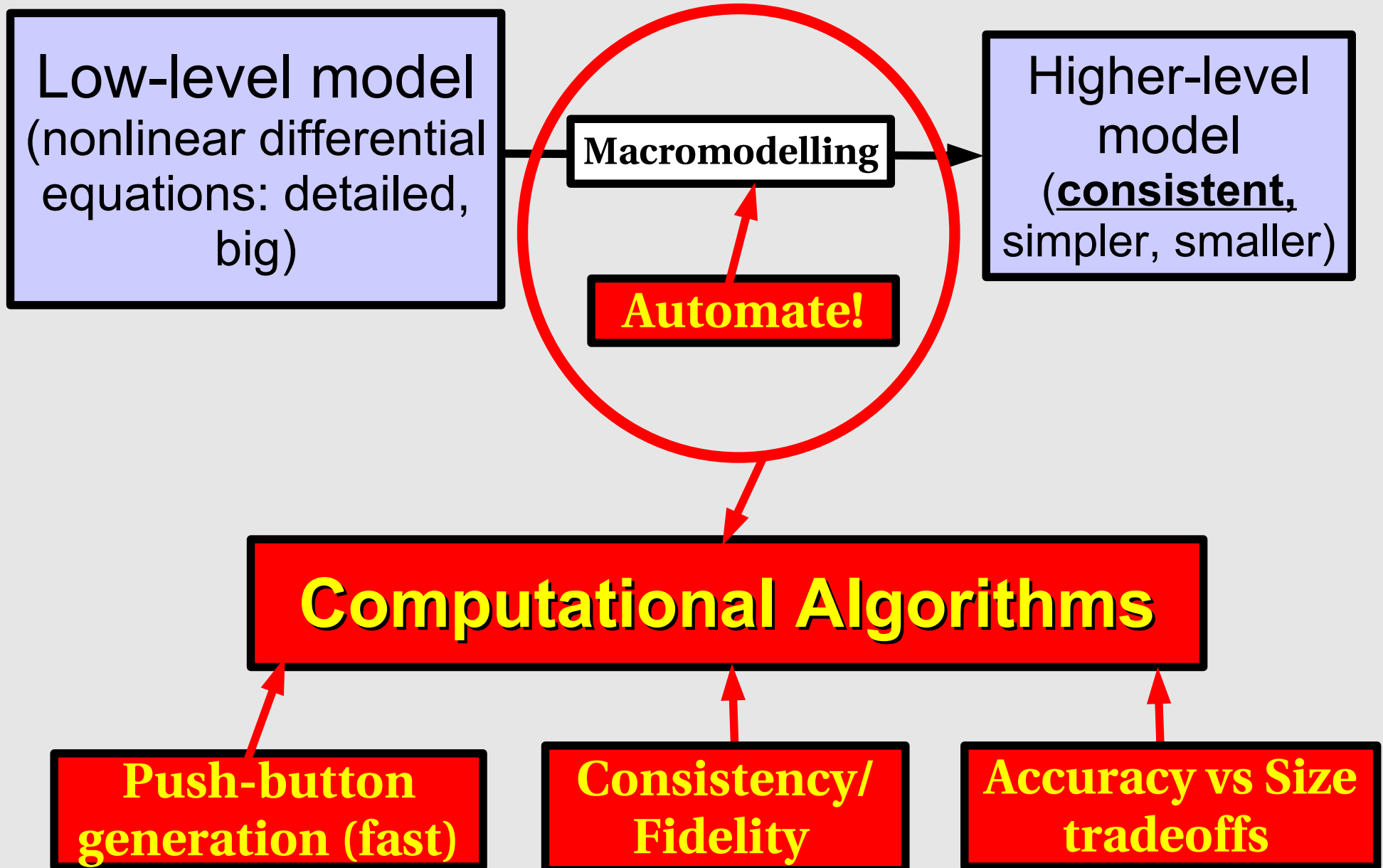
Low-level models



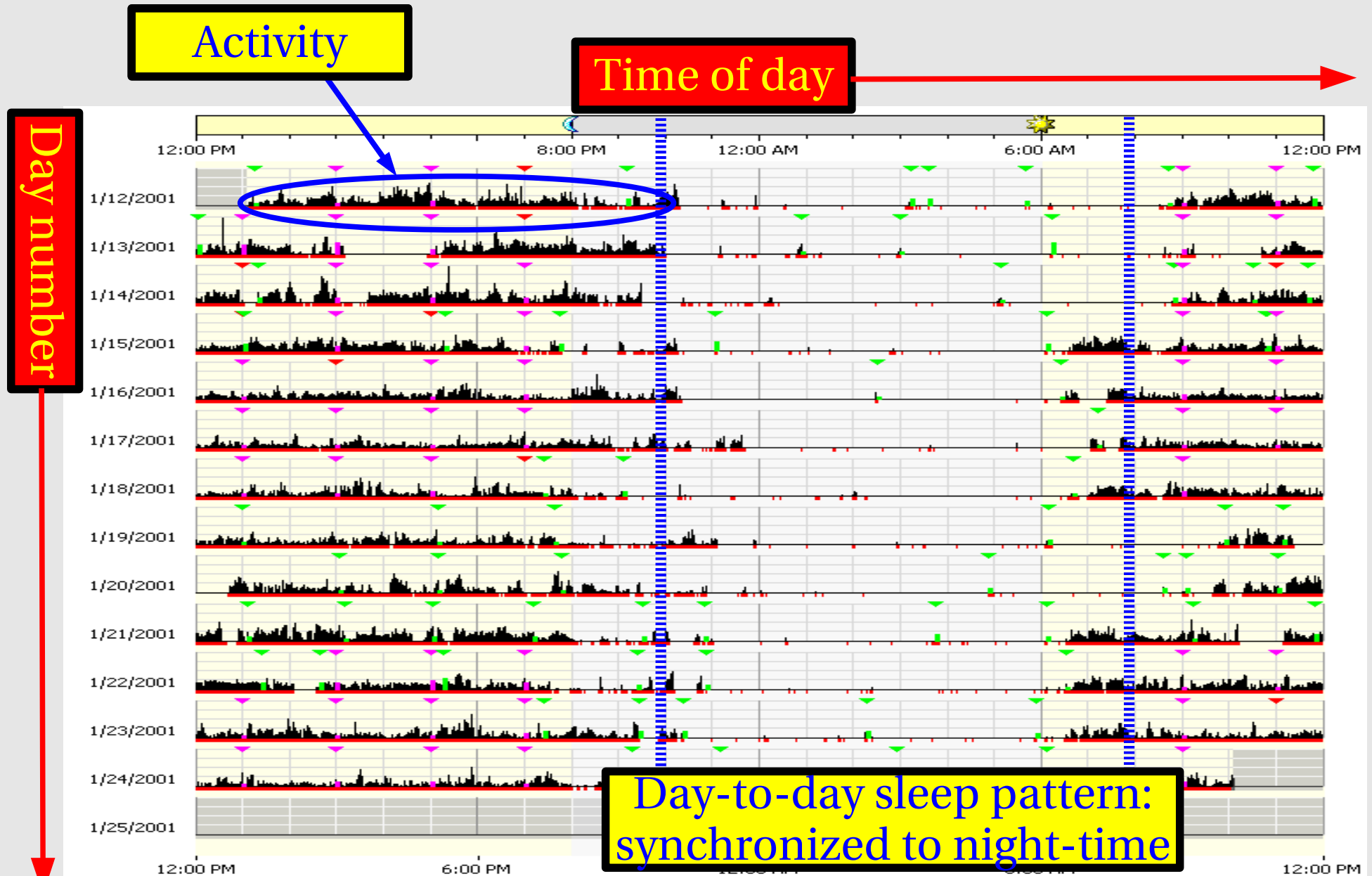
Model?

Accurate? Consistent?

# Computational Macromodelling



# Circadian Biological Oscillators (Human Sleep/Wake Patterns)

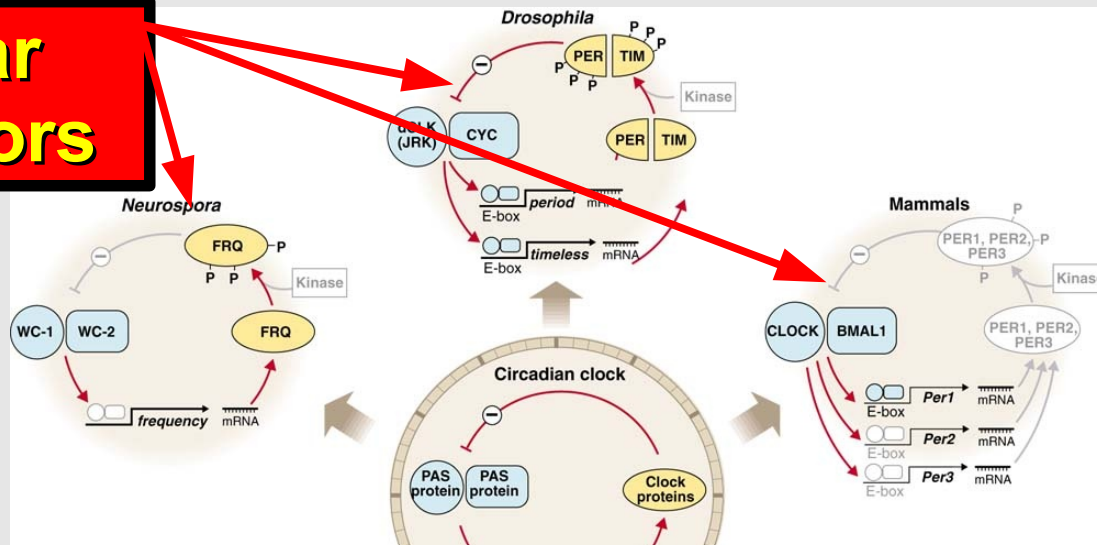


# Circadian Rhythm Oscillators



Credits: U Mass Amherst

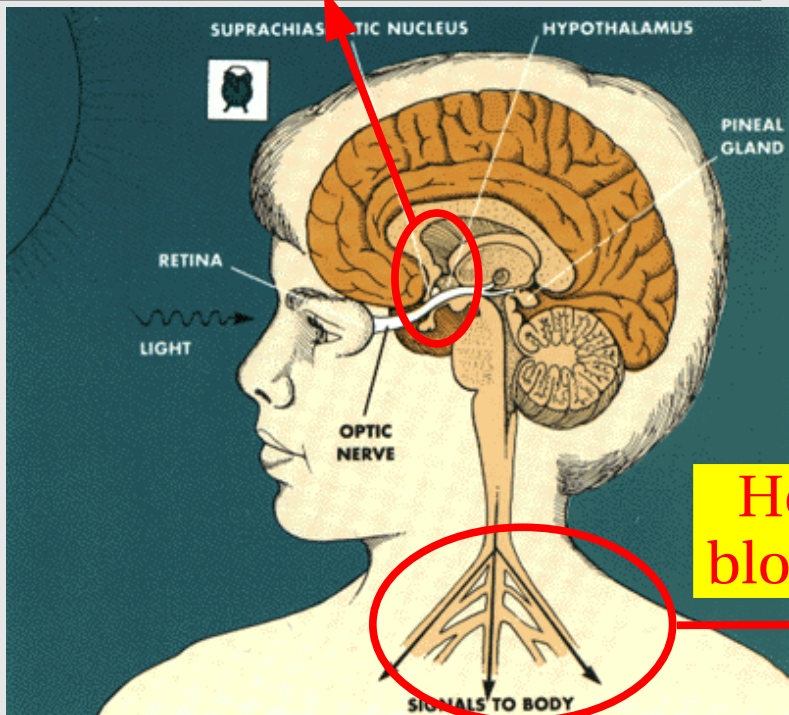
**Cellular Oscillators**



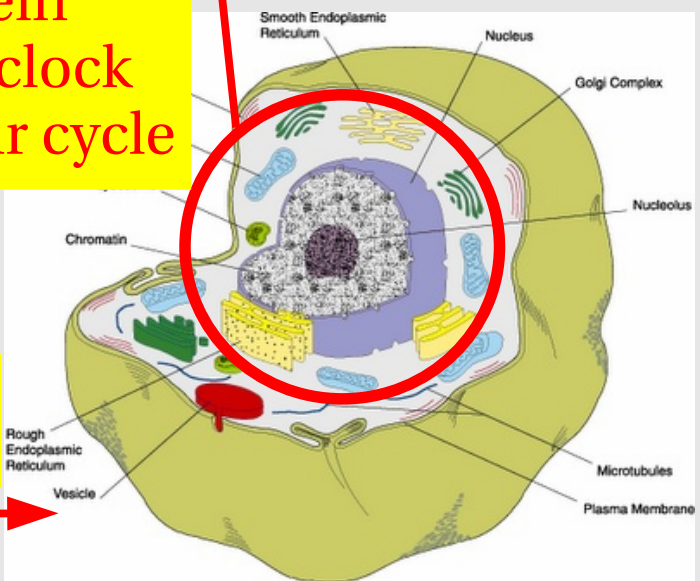
**Millions of interacting oscillators**

**Suprachiasmatic Nuclei (SCN)**  
lock to 24 hour period  
influence pineal gland and  
hypothalamus

**Intracellular DNA/protein biochemical clock locks to 24 hour cycle**

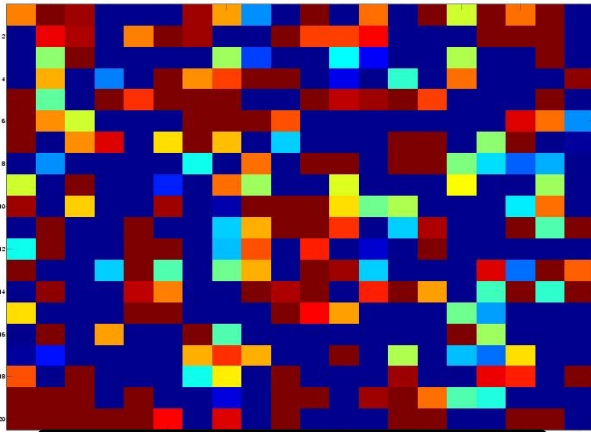


**Hormones through blood (eg, melatonin)**



Credits: scienceblogs.com

# Interacting Cellular Clocks

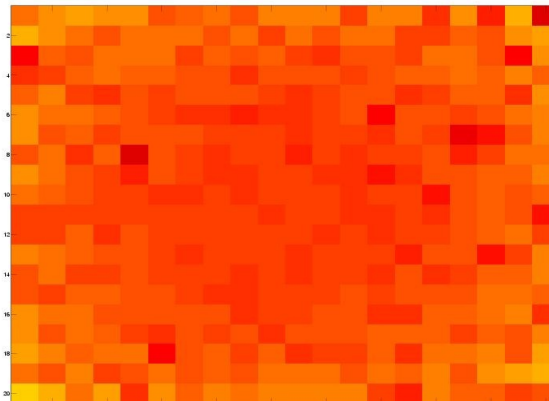


t=0



t=18h

- 16x16 grid of coupled cellular clocks



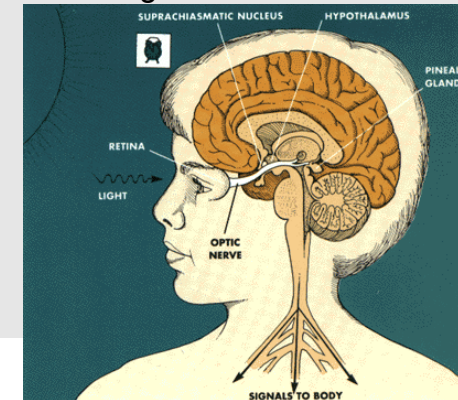
t=12h

Speedup: 240x

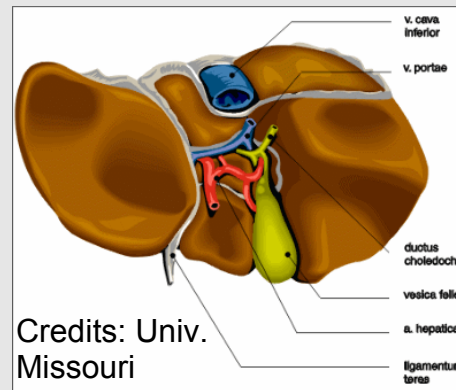
# Circadian Systems

## Organism

Credits: glimmerveen.nl

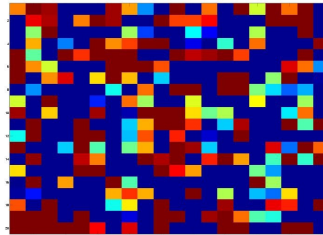


## Organ

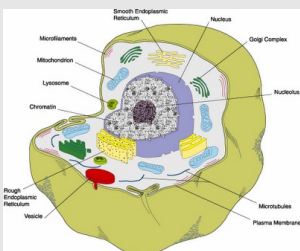


Credits: Univ. Missouri

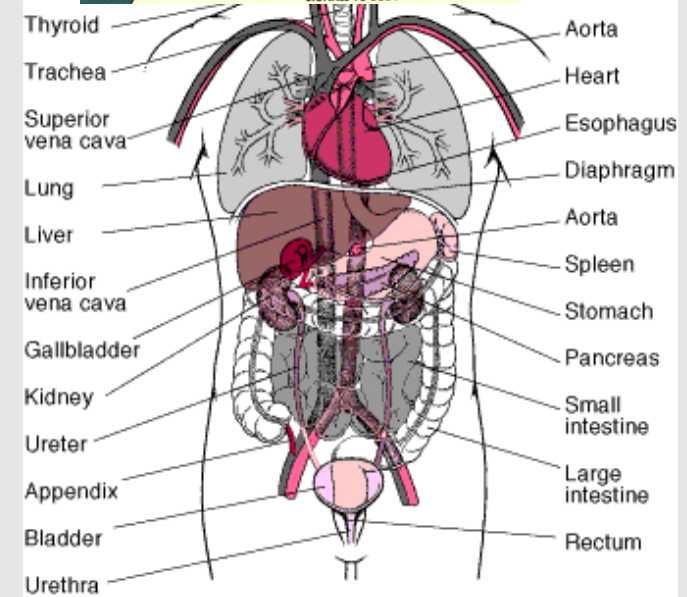
## Tissue



## Cell



Credits: scienceblogs.com



Credits: Iowa School for the Deaf

# Excerpt from SPICE3's *dioload.c*

```
#ifdef SENSDEBUG
    printf("vd = %.7e \n", vd);
#endif /* SENSDEBUG */
    goto next1;
}
if(ckt->CKTmode & MODEINITSMSIG) {
    vd= *(ckt->CKTstate0 + here->DIOvoltage);
} else if (ckt->CKTmode & MODEINITTRAN) {
    vd= *(ckt->CKTstate1 + here->DIOvoltage);
} else if ( (ckt->CKTmode & MODEINITJCT) &
            (ckt->CKTmode & MODETRANOP)
            && (ckt->CKTmode & MODEUIC) ) {
    vd=here->DIOinitCond;
} else if ( (ckt->CKTmode & MODEINITJCT) && here->DIOoff)
{
    vd=0;
} else if ( ckt->CKTmode & MODEINITJCT) {
    vd=here->DIOtVcrit;
} else if ( ckt->CKTmode & MODEINITFIX && here->DIOoff) {
    vd=0;
} else {
#ifdef PREDICTOR
    if (ckt->CKTmode & MODEINITPRED) {
```

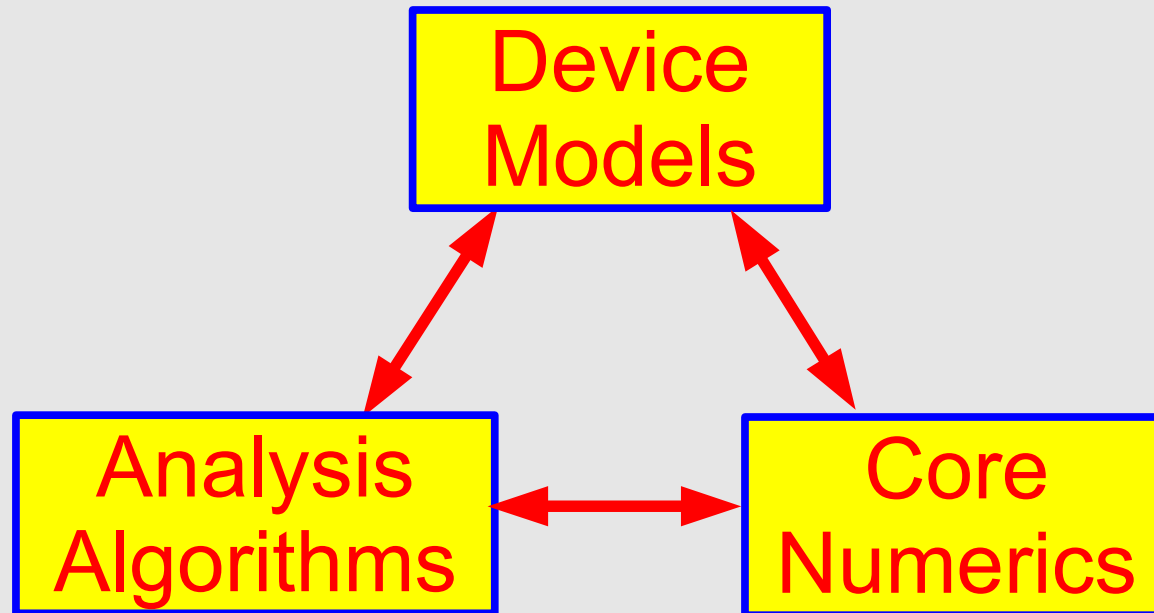
Sensitivity analysis code

AC analysis code

Transient analysis related code

PREDICTOR

# Modular Software Infrastructure for Research in Simulation



- foster collaborative research, effective teaching
  - modular (reflecting mathematical structure)
  - open source, freely distributed, effectively documented
    - reproducible research
  - short, high-level code (MATLAB/Octave/Python)