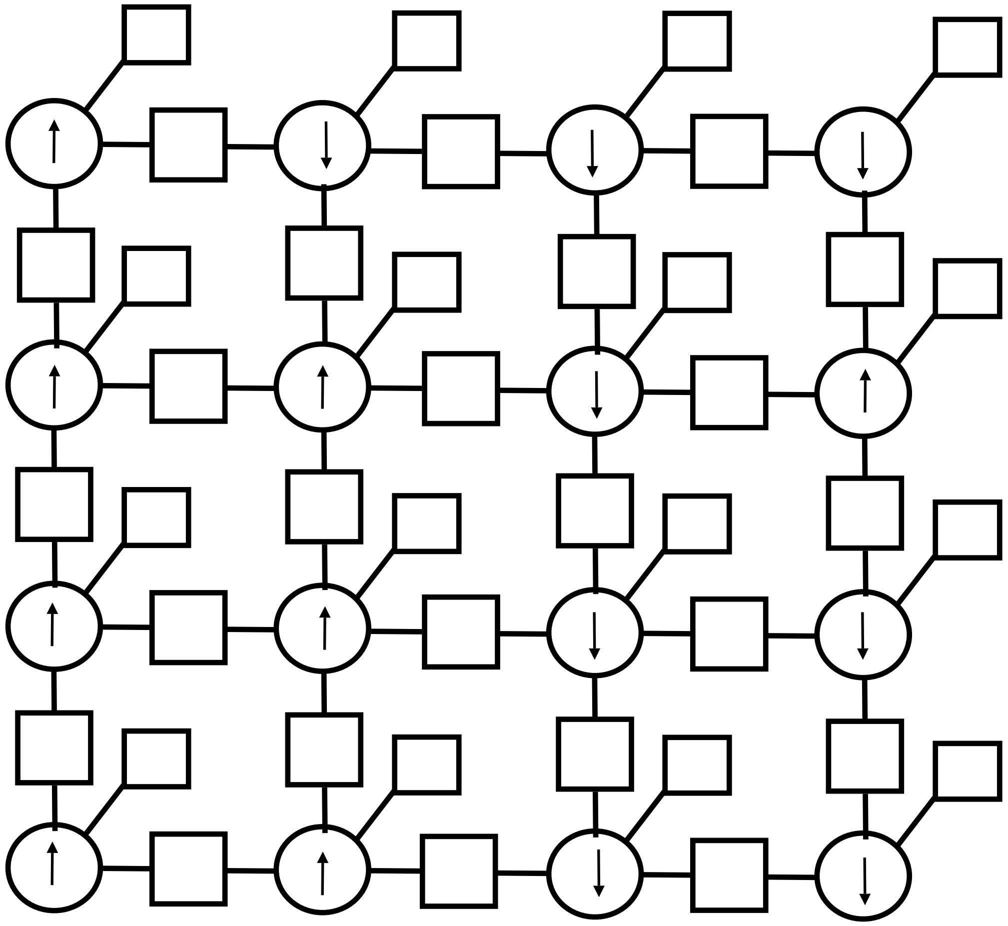
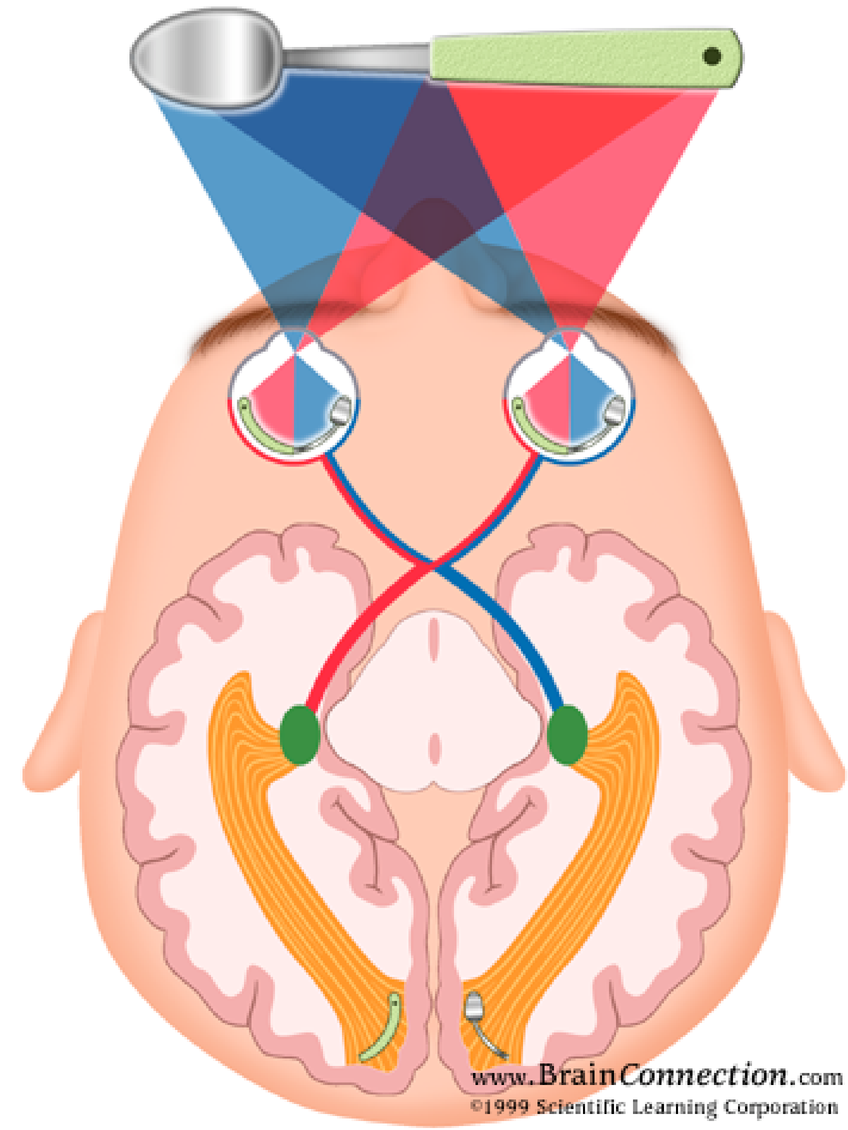
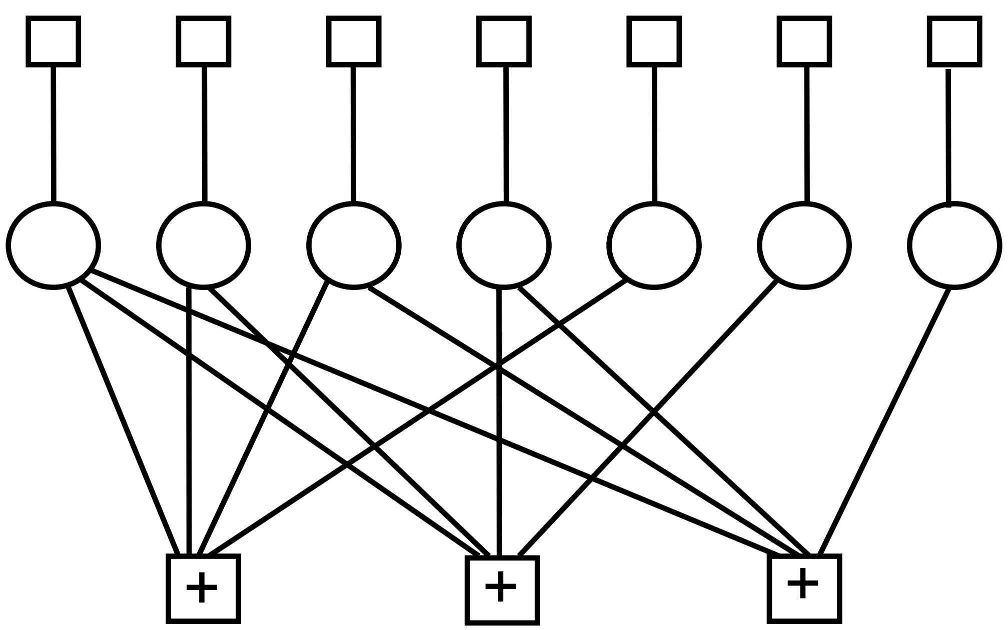
Multi-cellular Logic Circuits

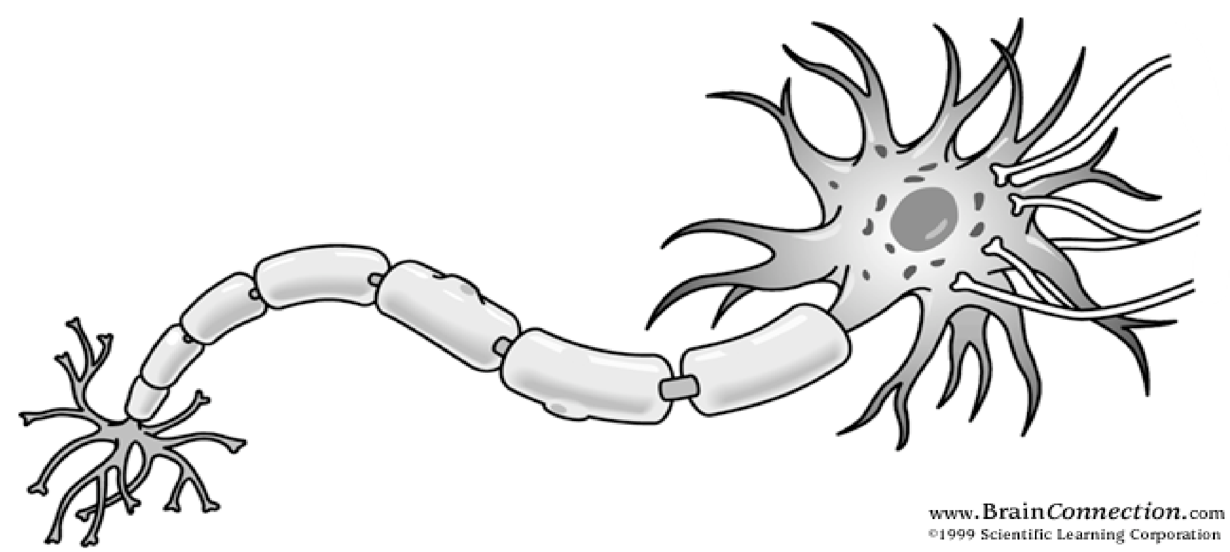
Jonathan Yedidia

Mitsubishi Electric Research Laboratories




www.BrainConnection.com
©1999 Scientific Learning Corporation

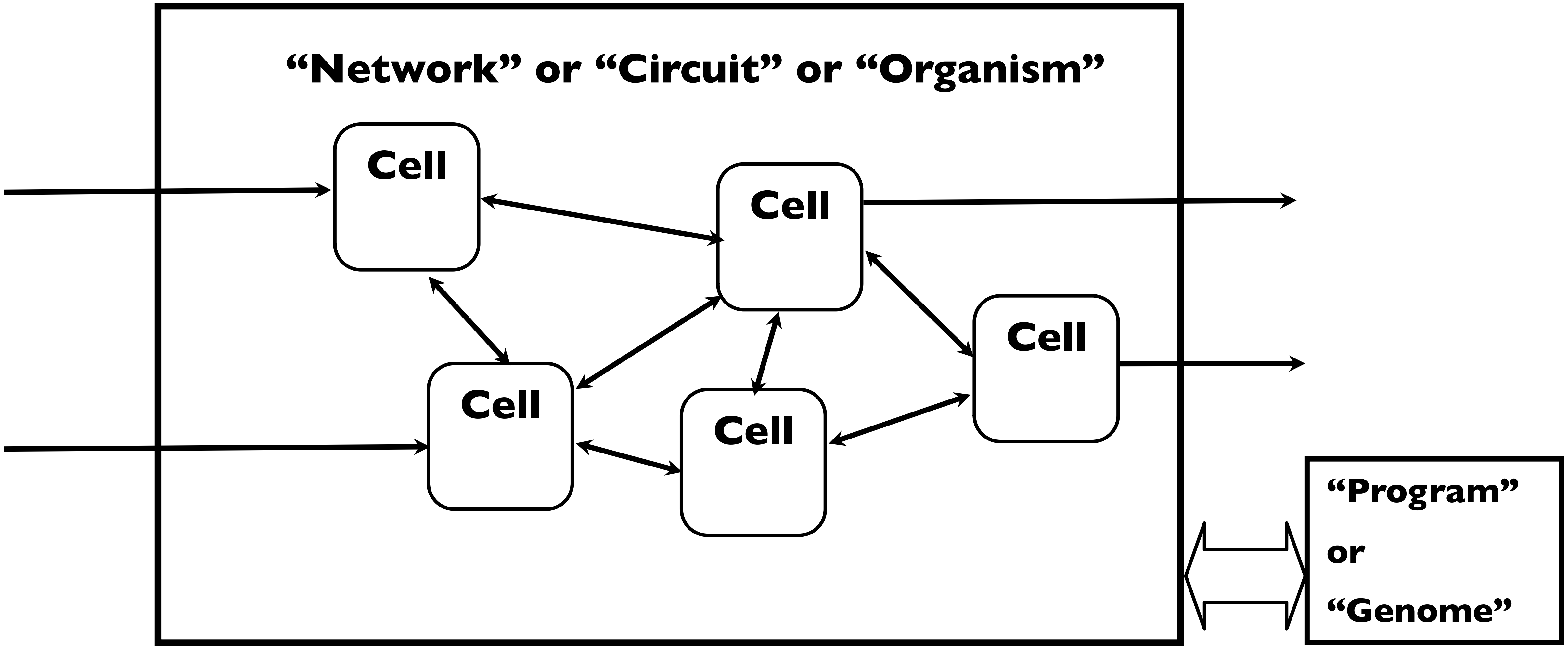



www.BrainConnection.com
©1999 Scientific Learning Corporation

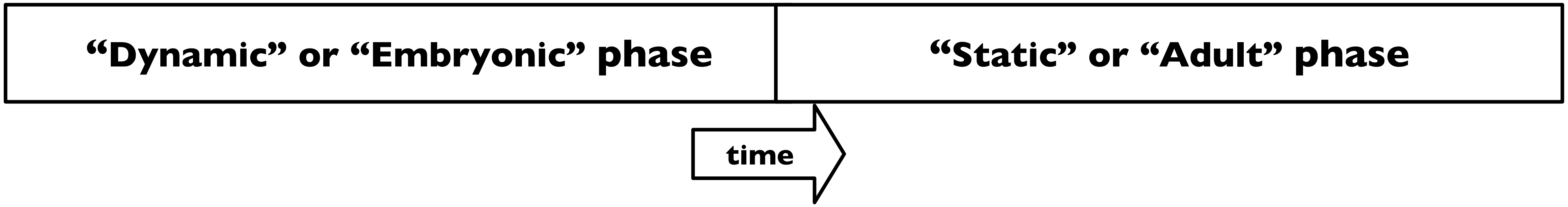
Outline

- A model of the logic of multi-cellular organisms
- Examples of the design strategy:
  - a RAM circuit
  - a weighted-least-squares smoothing circuit
- Biology background
- Discussion and future prospects

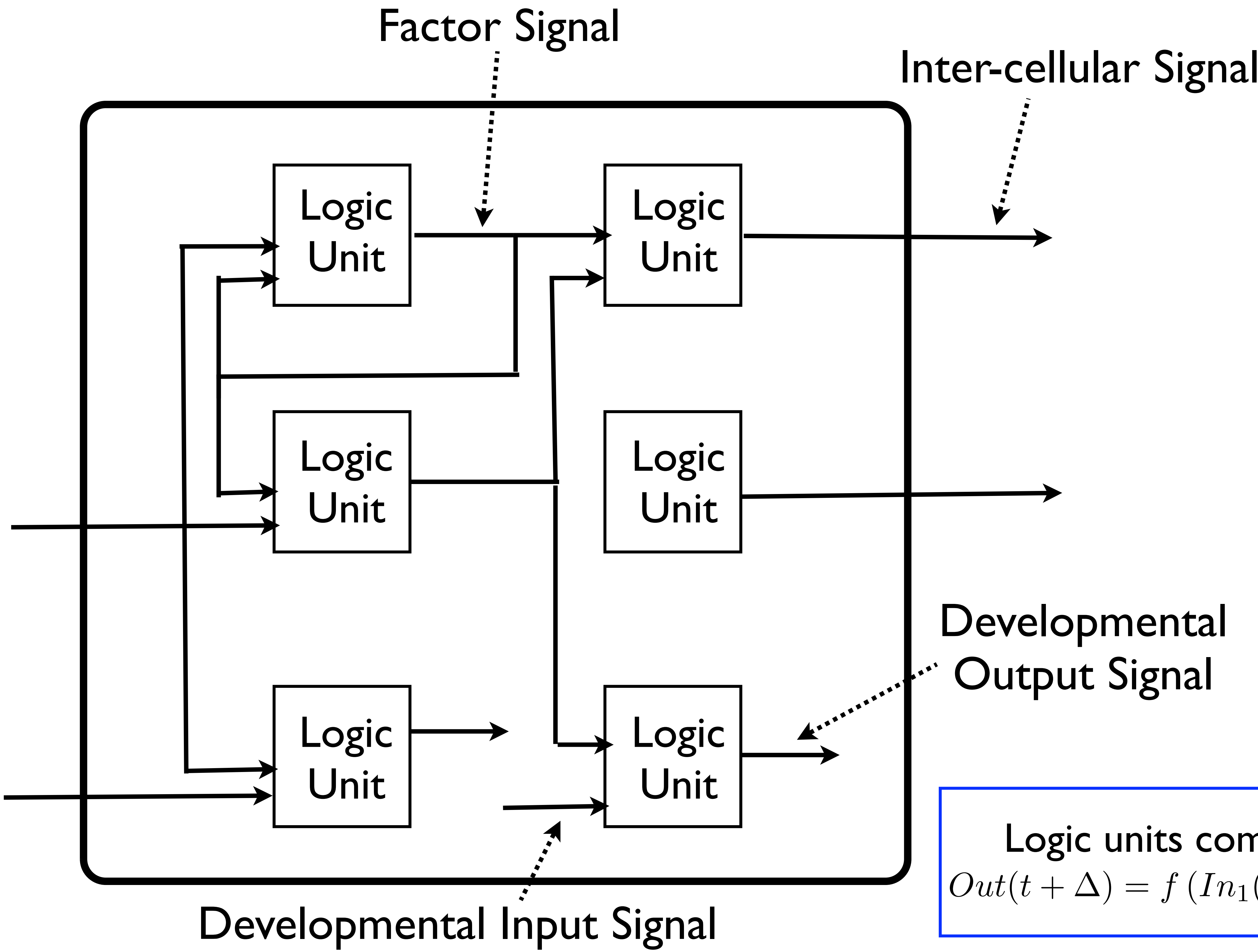
A Simple Model



Key point: all cells have identical specifications



Inside a Cell

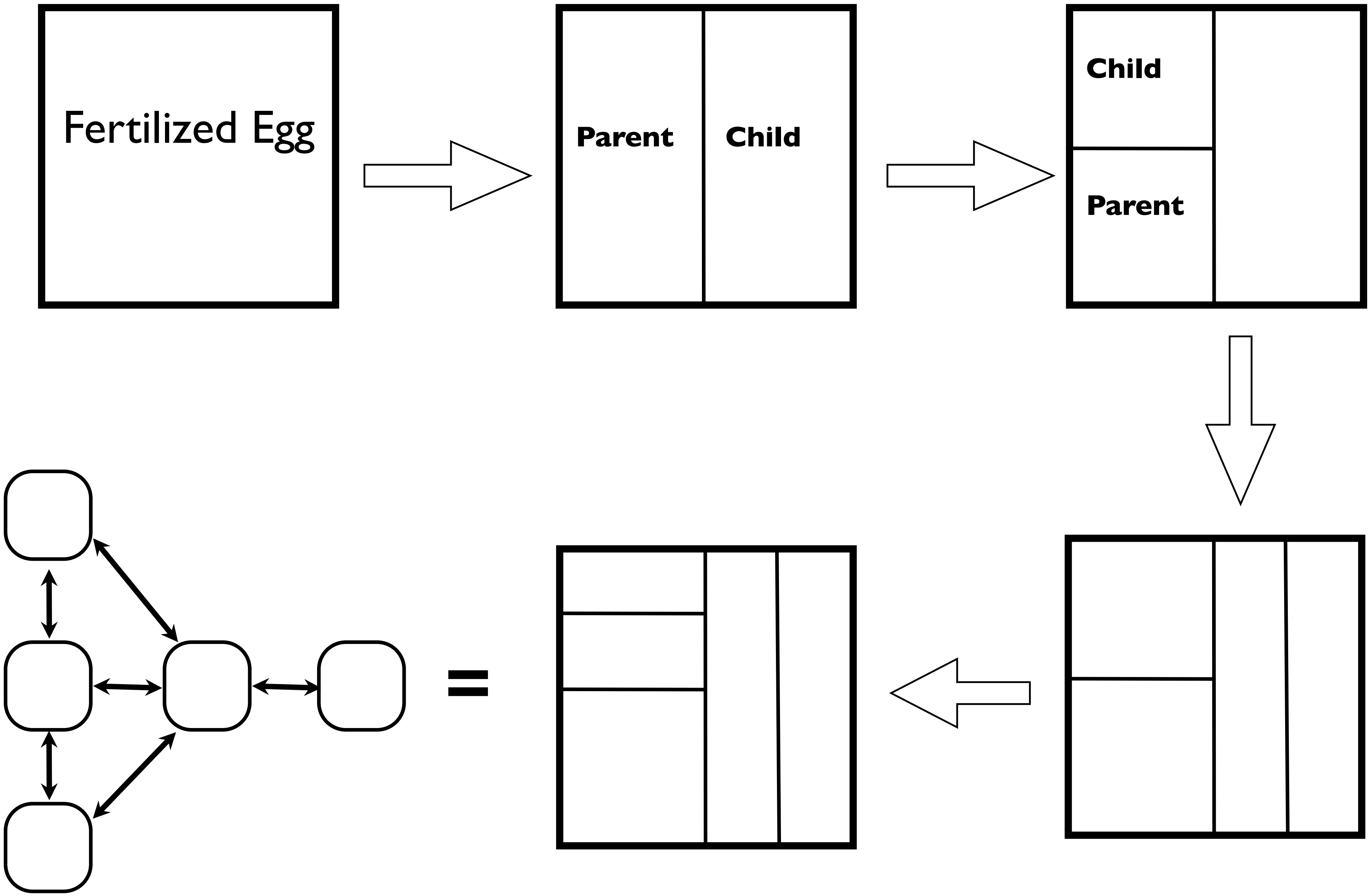


Logic units compute:

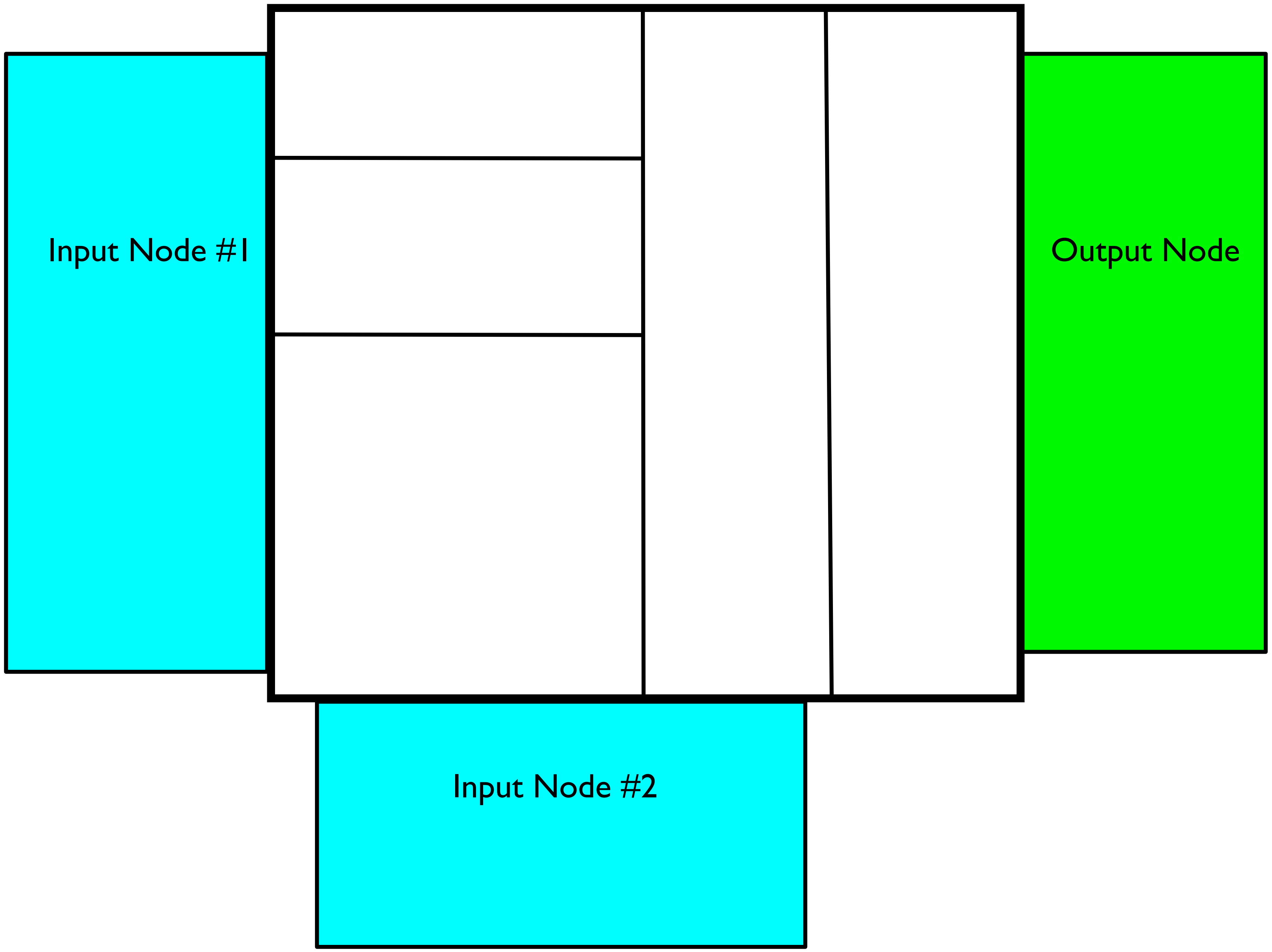
$$Out(t+\Delta) = f\left(In_1(t), ..., In_k(t)\right)$$

Developmental Phase

Cell Divisions Triggered by "Developmental Output Signals"



Testing the Adult Circuit



More Details and Embellishments

- Reconciliation Functions: If a two or more logic units produce the same output signal, or if a cell receives the same inter-cellular signal from two or more neighboring cells, the value of the signals must be reconciled. For binary signals, an OR function is used.

- Digital, analog, or mixed signals can be used. If analog signals are used, it might make sense to keep them in the range [0, 1], and to use soft logic functions like

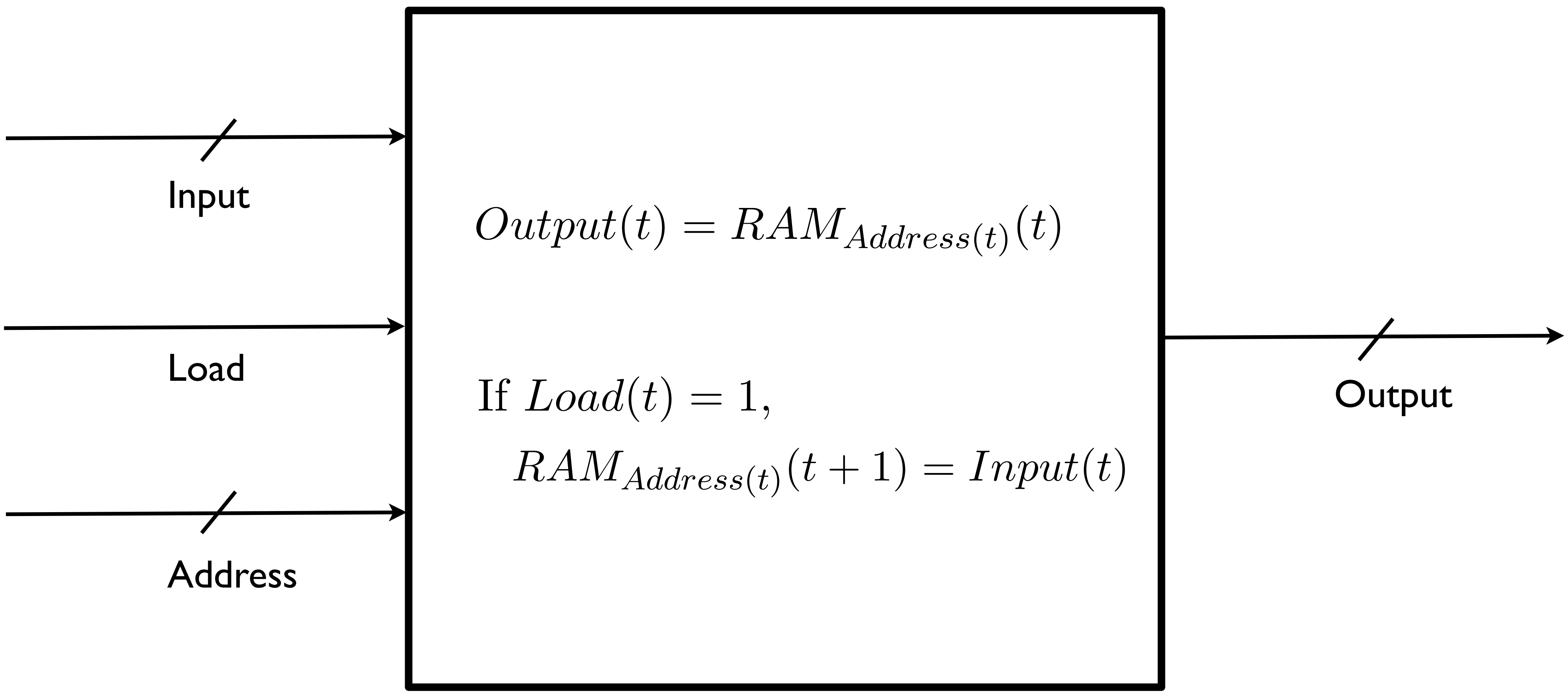
$$Out(t + \Delta) = f(In_1 + In_2 - In_1 In_2)$$

$$f(x) = \frac{x^k}{x^k + (1-x)^k}$$

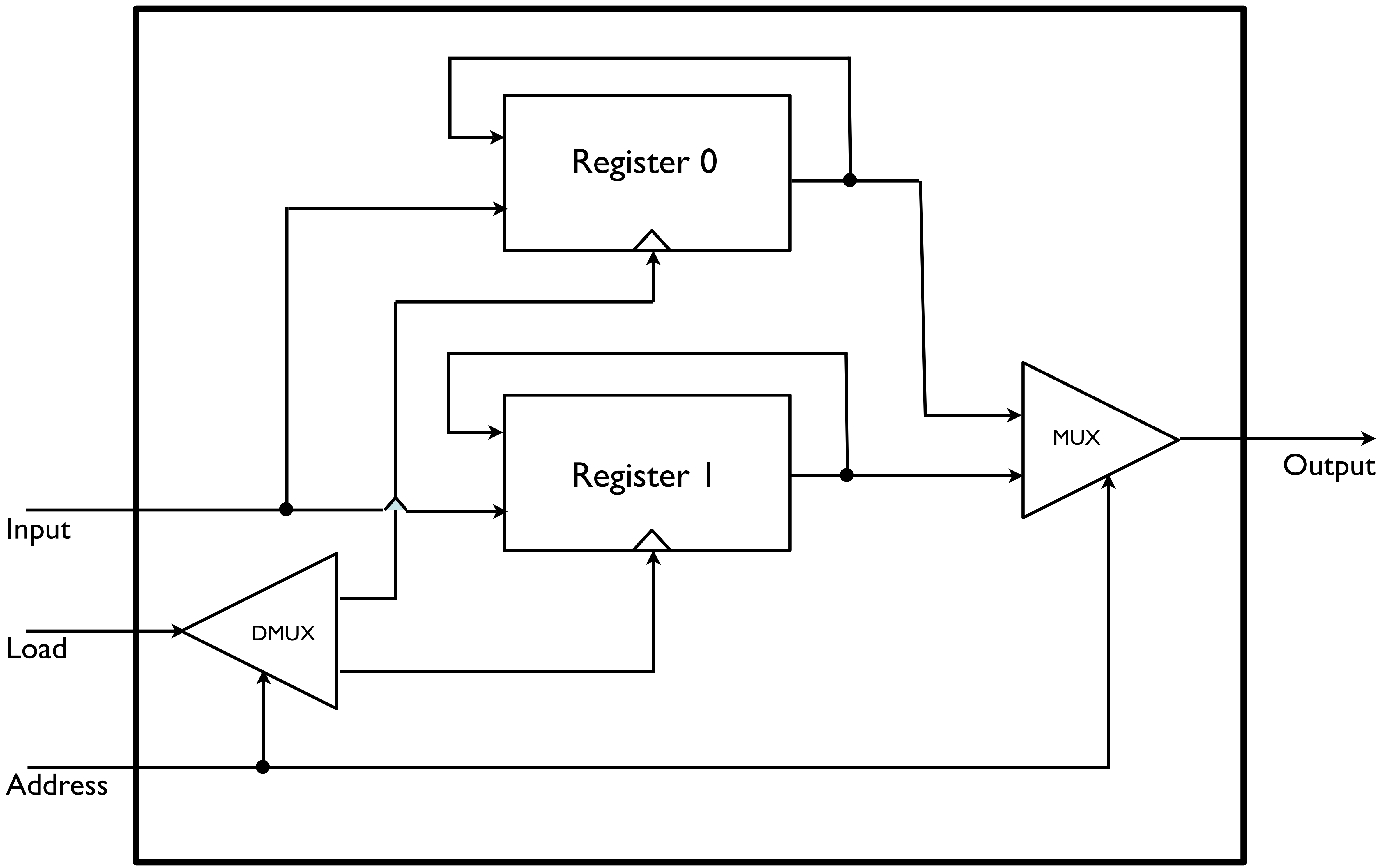
- Logic units can be synchronized to a global clock that has discrete time steps; or the cells can be asynchronous, and the logic units could have arbitrary delays.

- Sometimes it's useful to include "receptors" which only allow inter-cellular signals of certain types to enter a cell from certain directions.

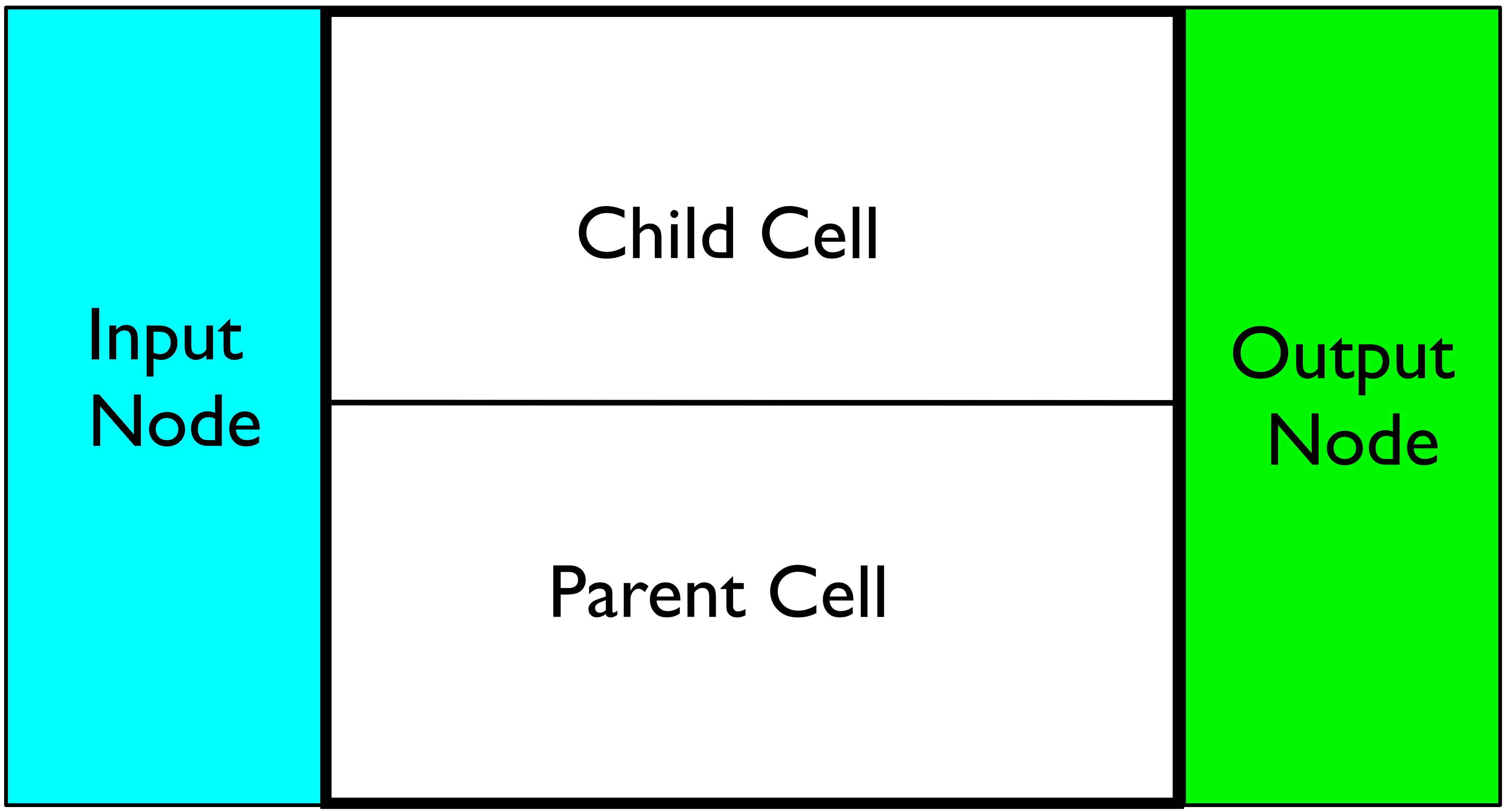
A Random-Access Memory



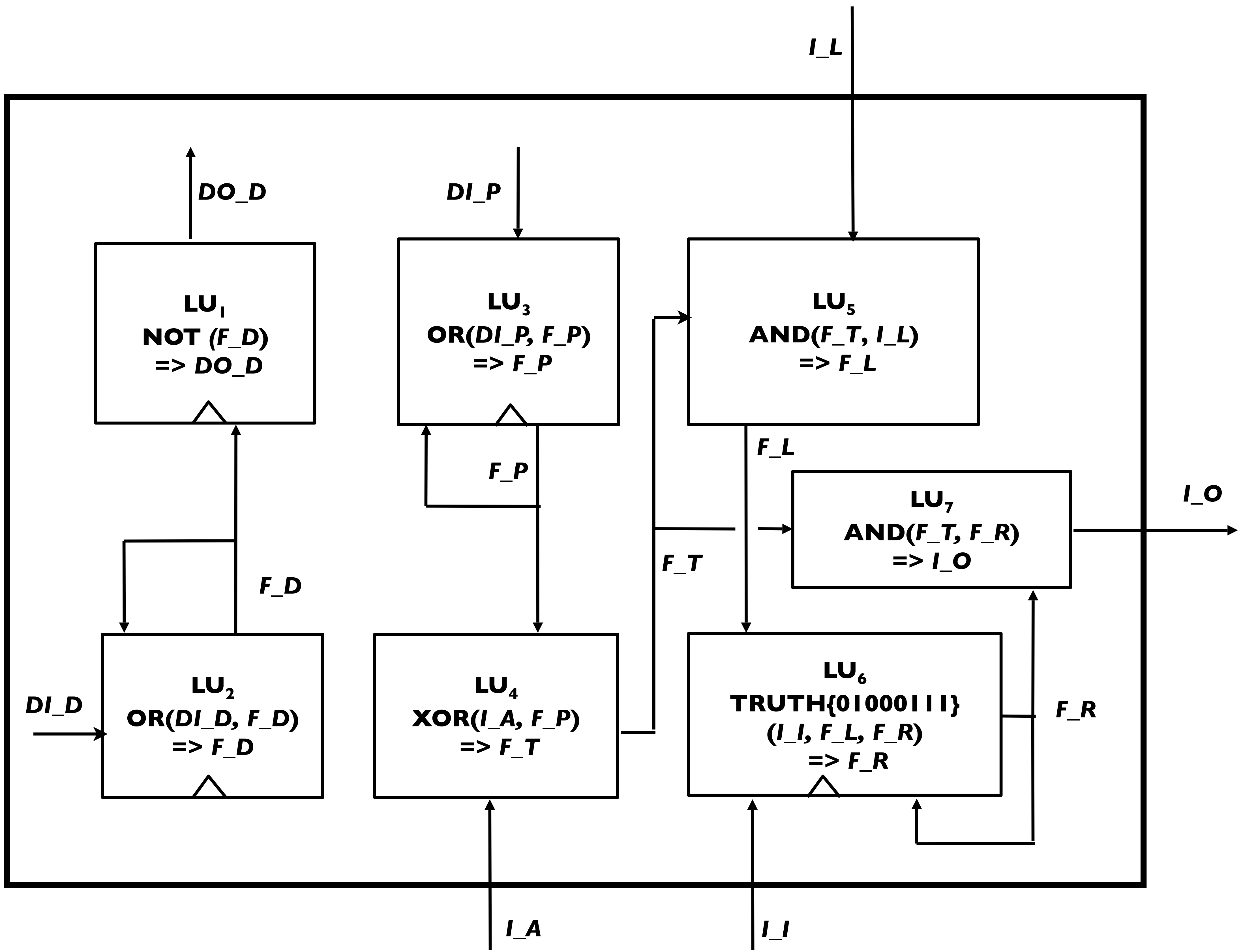
Standard Design (for Tiny-RAM)



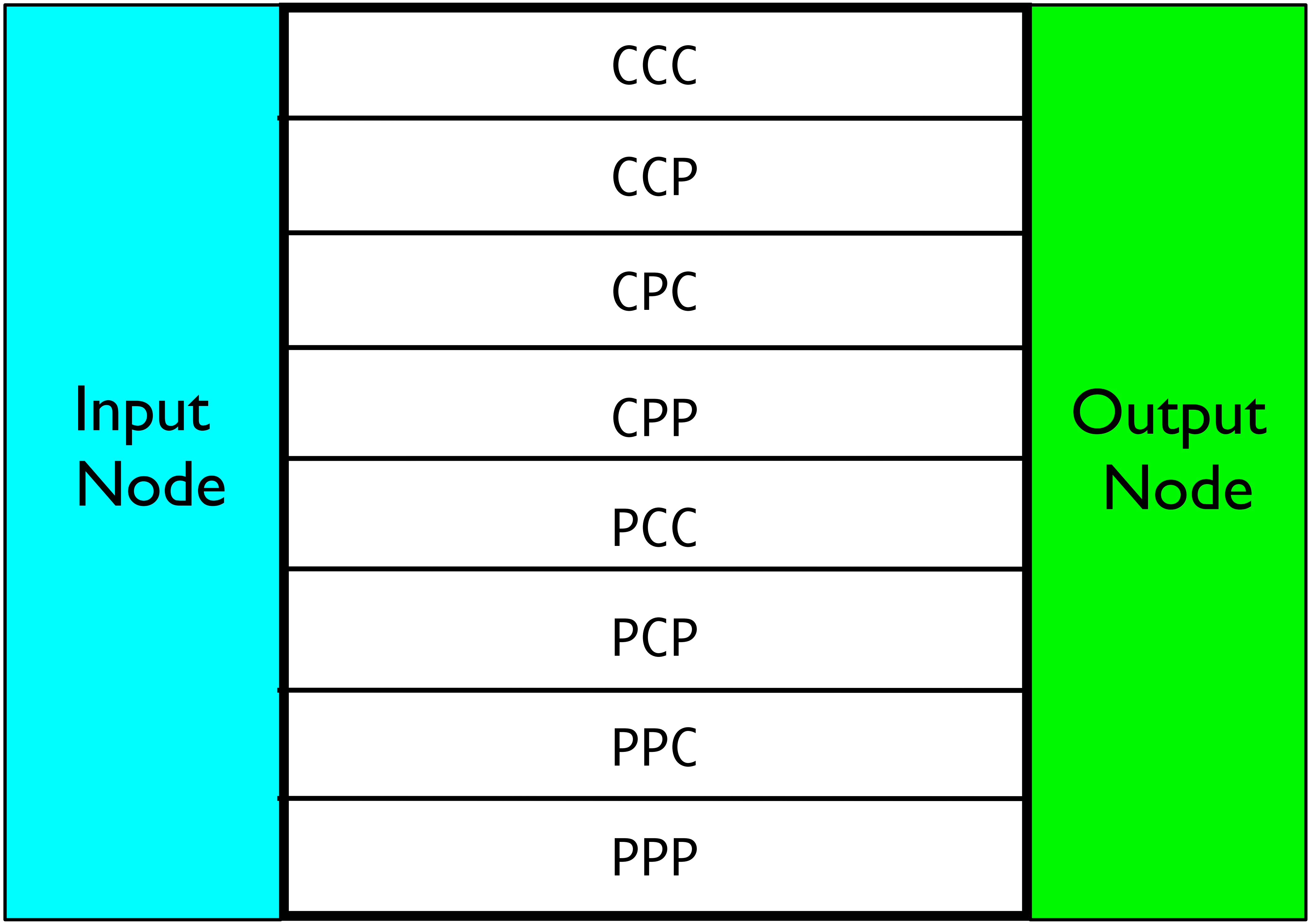
Set-up for Adult Tiny-RAM



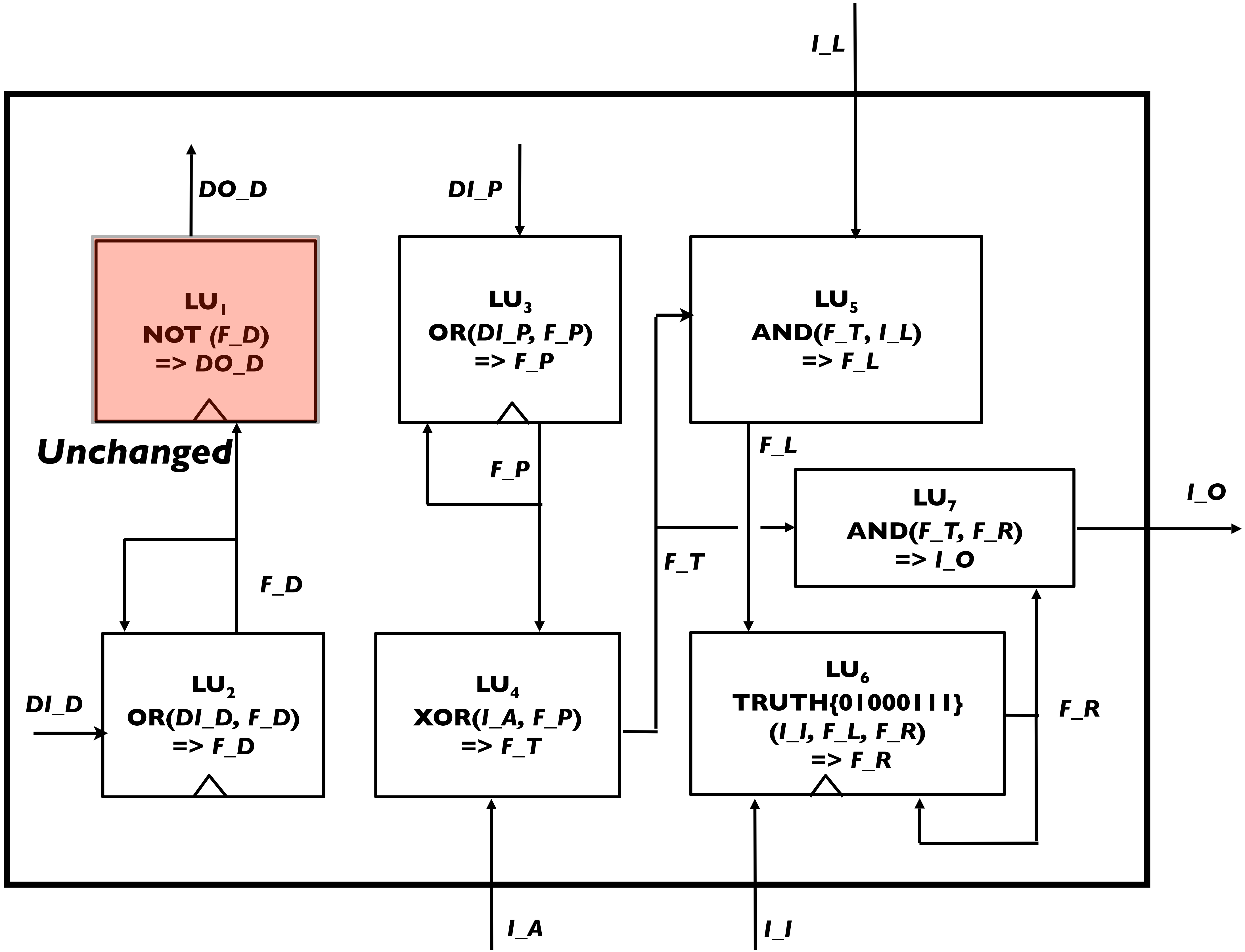
Logic Units in One Tiny-RAM Cell



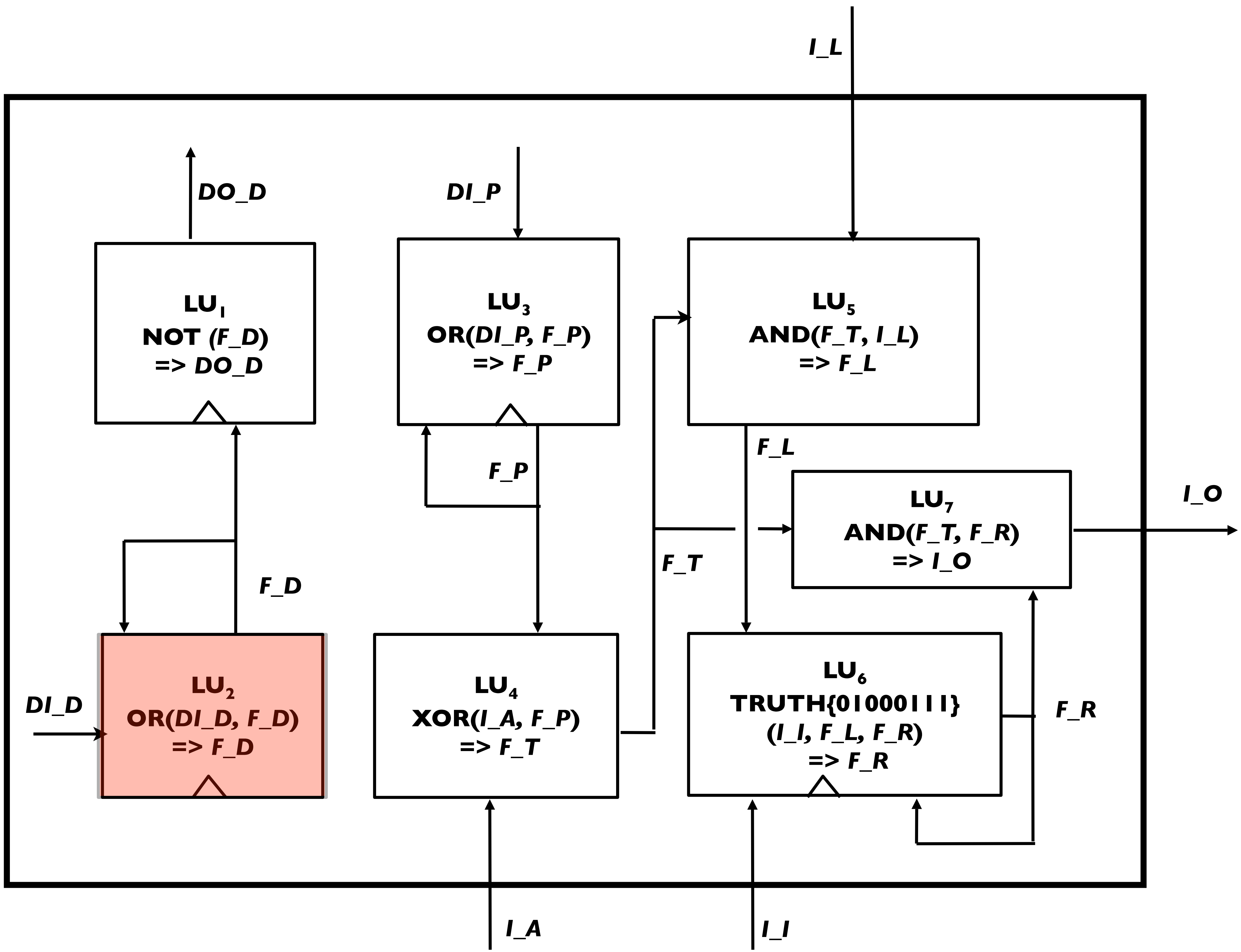
Scaling up the RAM



Scaling Changes



Scaling Changes

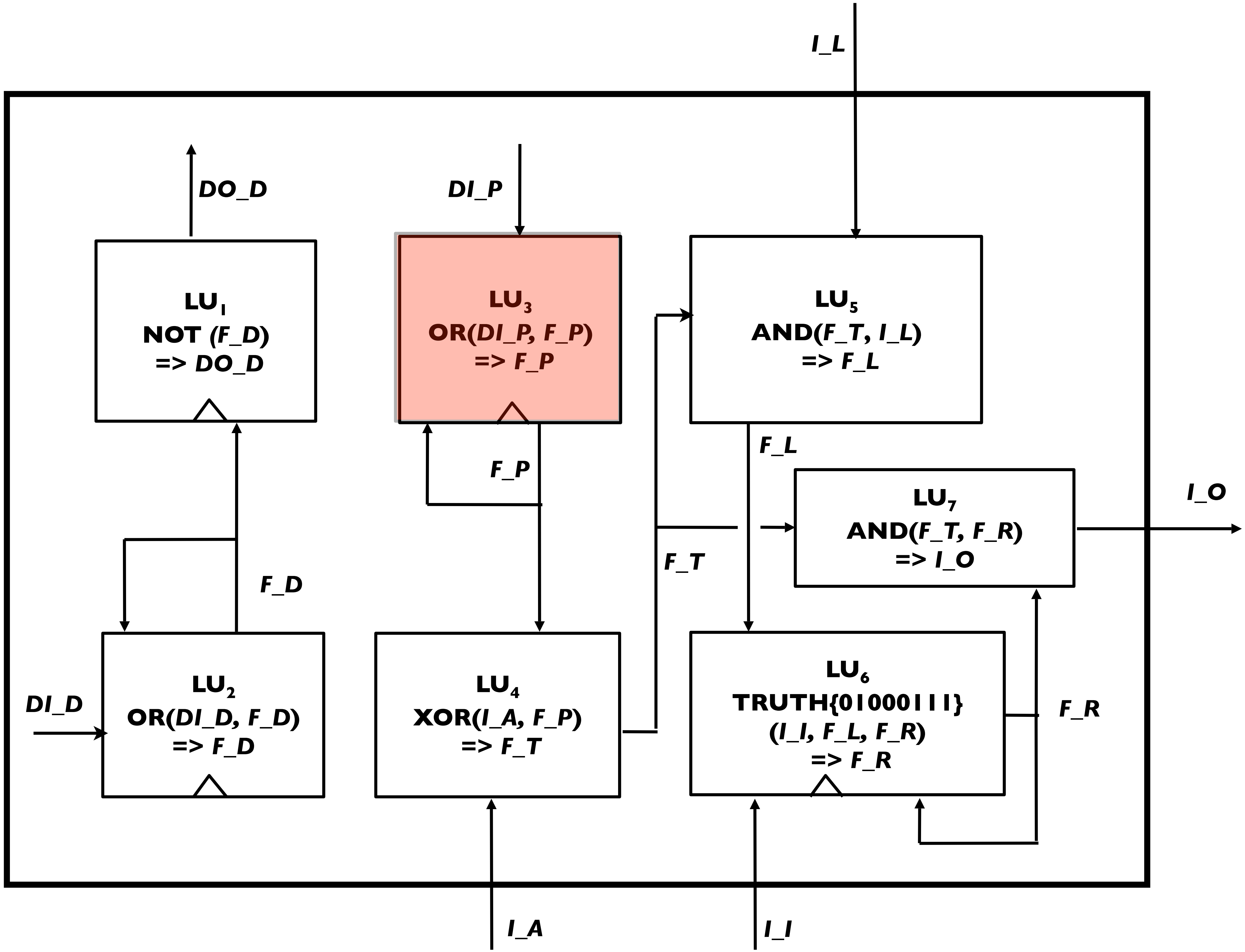


LU2: Or(DI_D, F_D) => F_D

replaced with

- Or(DI_D, F_D1) => F_D1
- Or(And(DI_D, F_D1), F_D2) => F_D2
- ...
- Or(And(DI_D, F_D[K-1]), F_D) => F_D

Scaling Changes

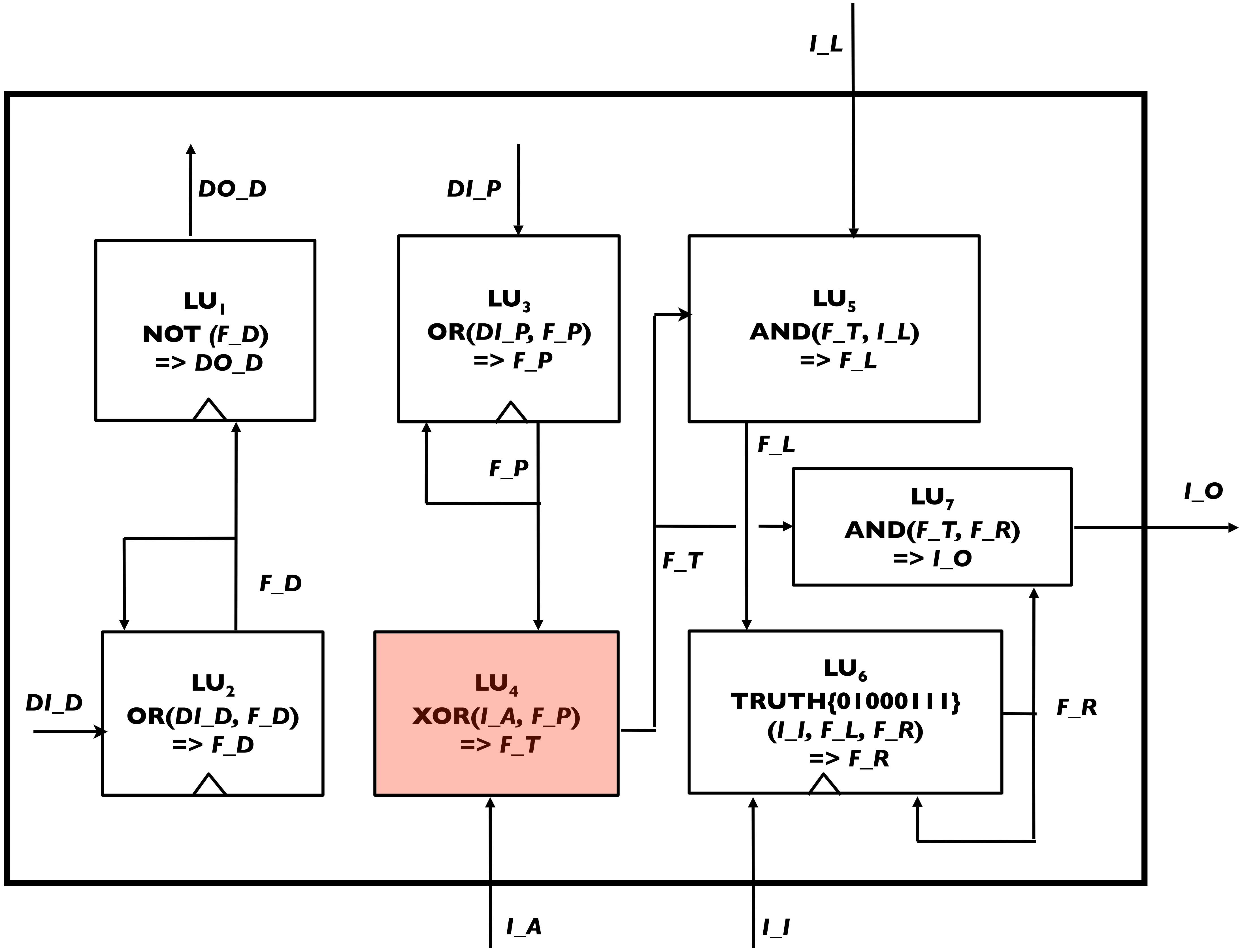


LU3: Or(DI_P, F_P) => F_P

replaced with

- Or(And(DI_P, Not(F_D1)), F_P1) => F_P1
- Or(And(DI_P, F_D1, Not(F_D2)), F_P2) => F_P2
- ....
- Or(And(DI_P, F_D[k-1], Not(F_D)), F_P[K]) => F_P[K]

Scaling Changes



LU4: XOR(I_A, F_P) => F_T

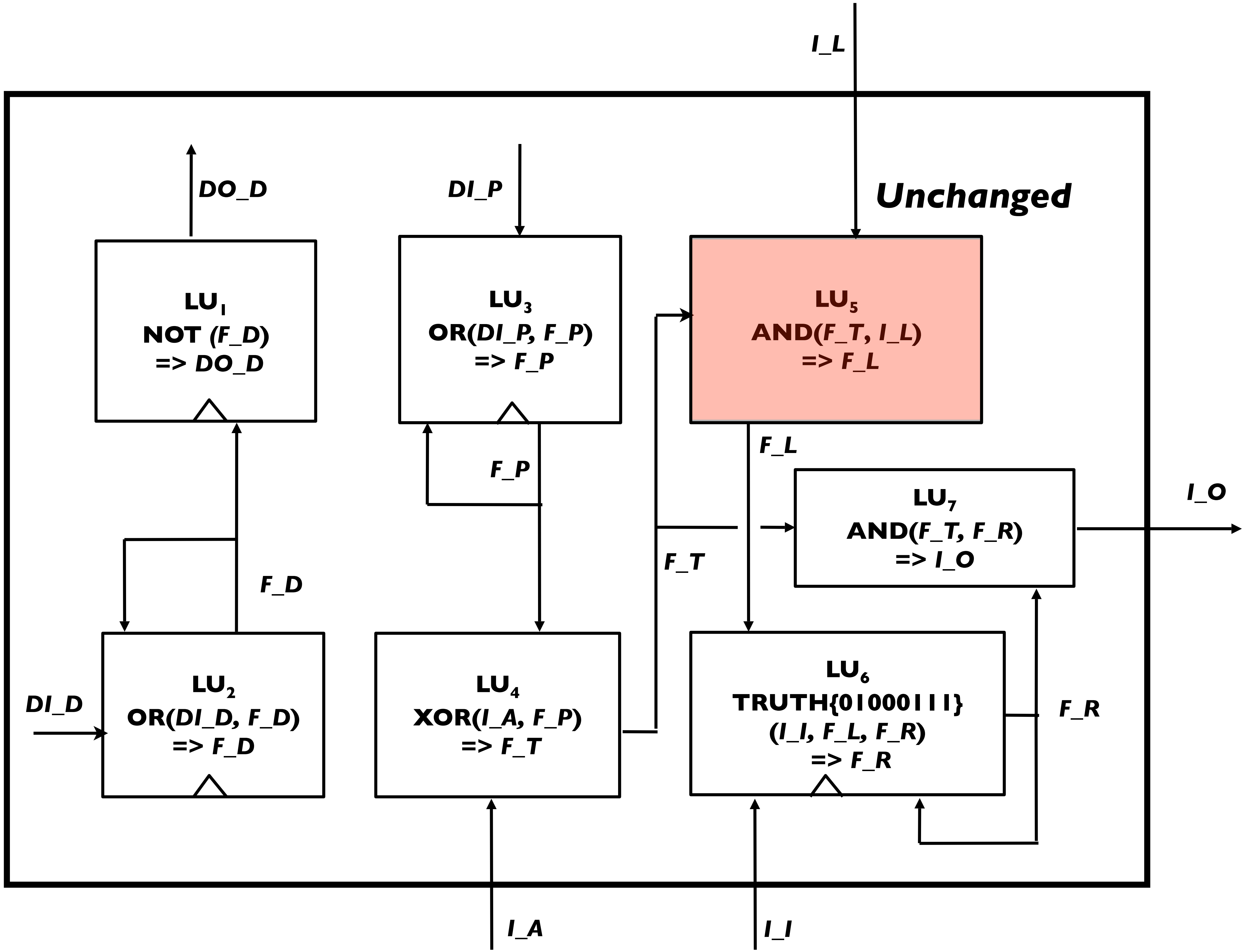
replaced with

- XOR(I_A1, F_P1) => F_T1
- XOR(I_A2, F_P2) => F_T2
- ....
- XOR(I_A[k], F_P[k]) => F_T[K]

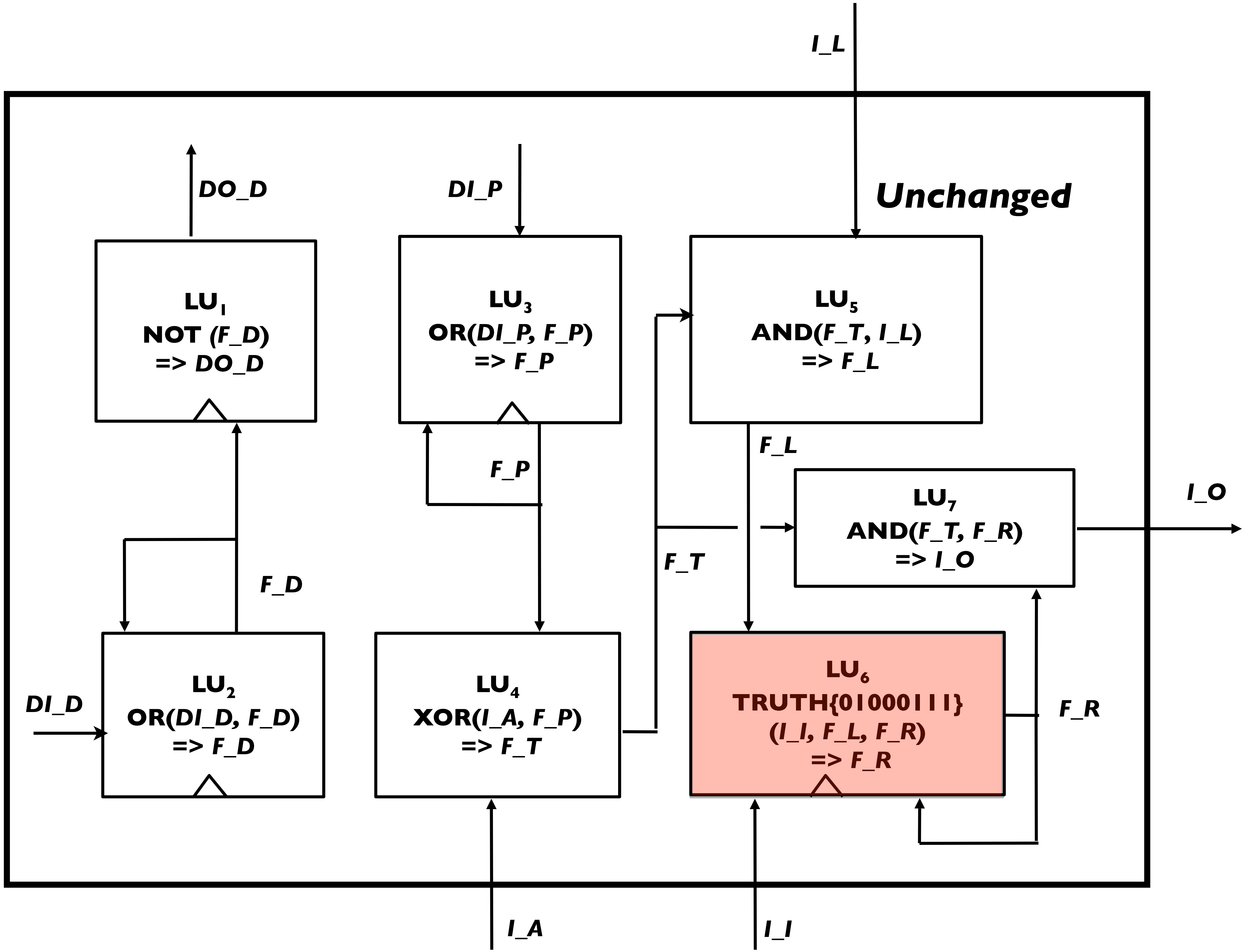
plus

- And(F_T1, F_T2, ... , F_T[K]) => F_T

Scaling Changes



Scaling Changes

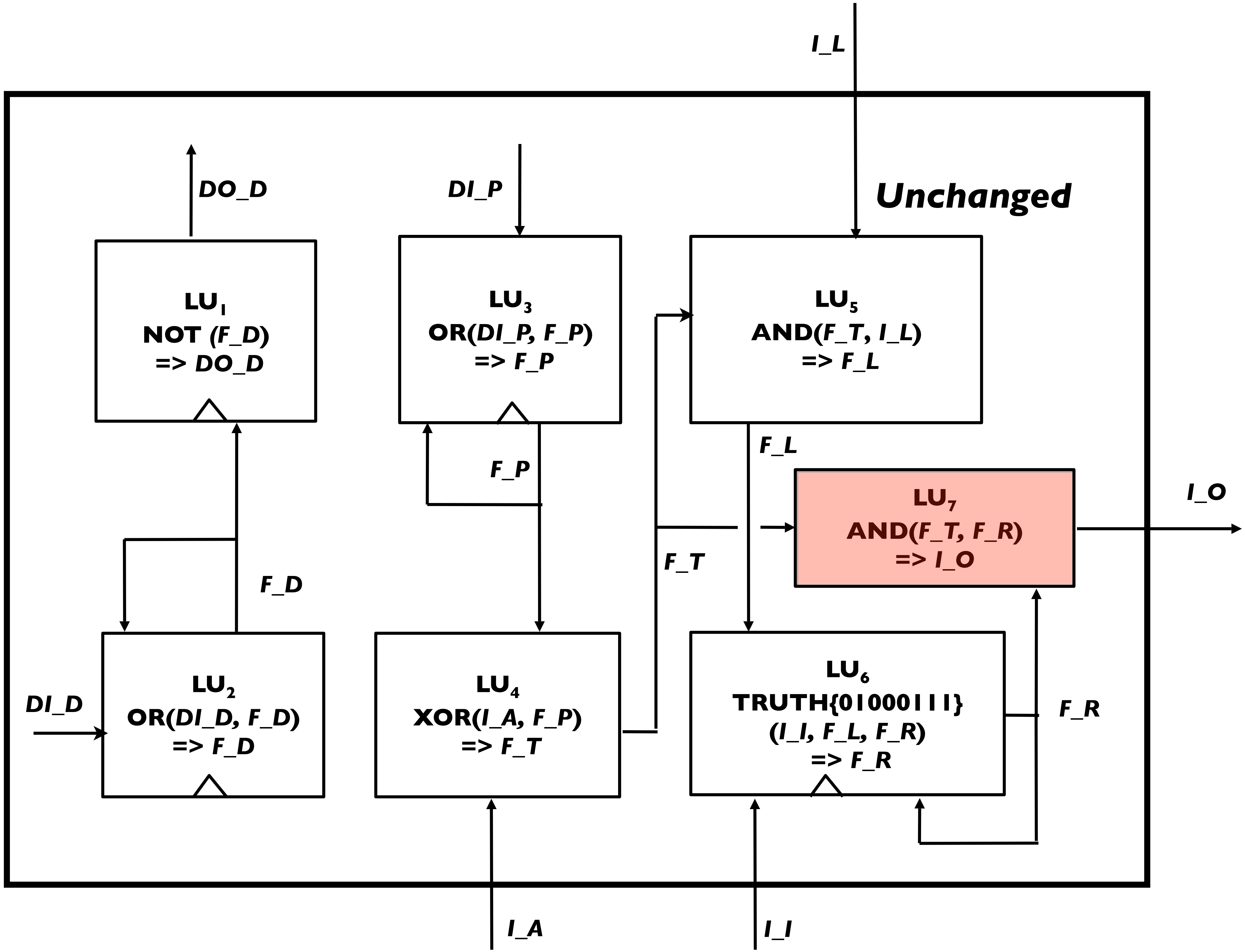


LU6: TRUTH{01000111}(I_I, F_L, F_R) => F_R

replaced with

- TRUTH{01000111}(I_I1, F_L, F_R1) => F_R1
- TRUTH{01000111}(I_I2, F_L, F_R2) => F_R2
- ....
- TRUTH{01000111}(I_I[L], F_L, F_R[L]) => F_R[L]

Scaling Changes

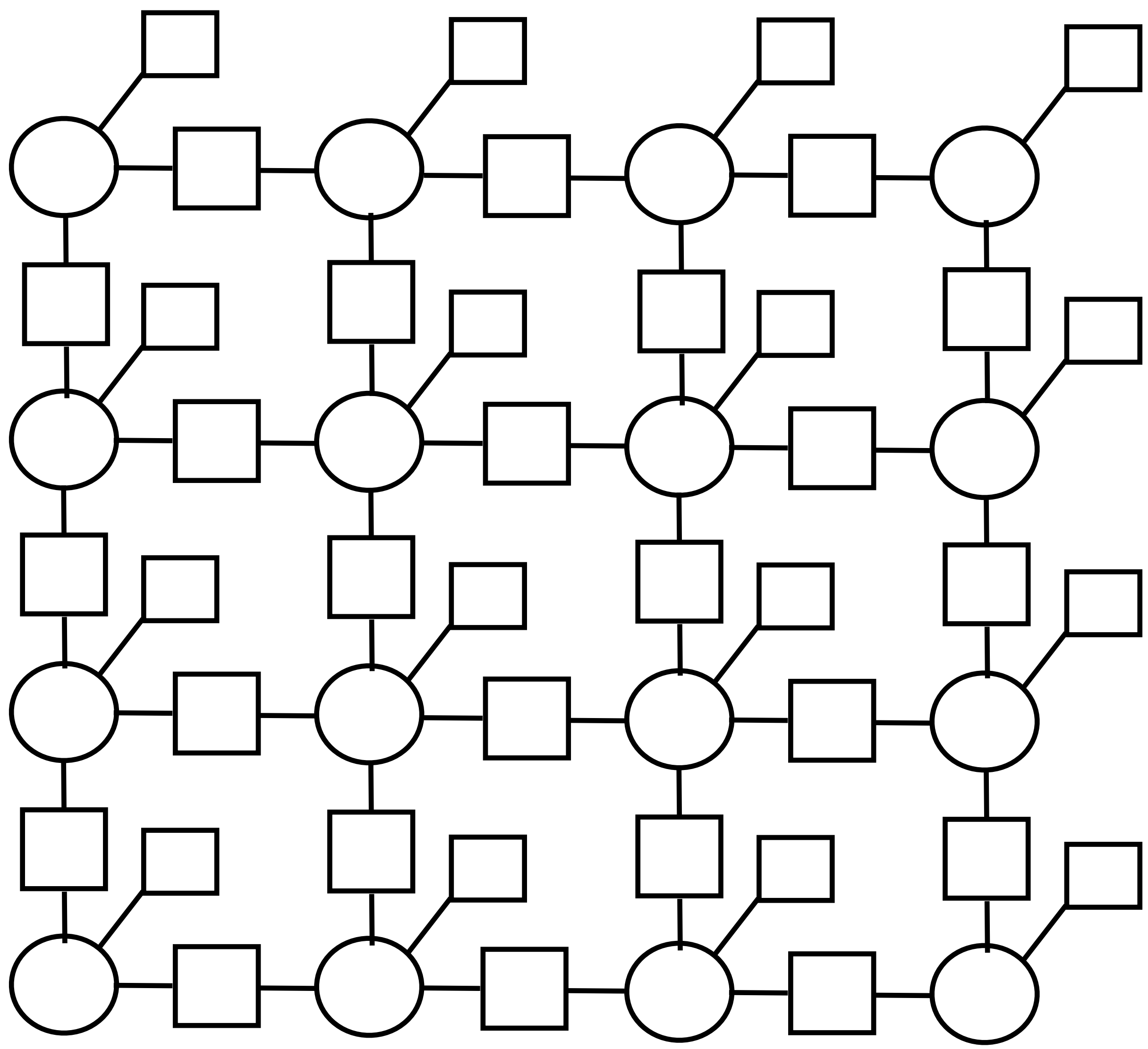


LU7: And(F_T, F_R) => I_O

replaced with

- And(F_T, F_R1) => I_O1
- And(F_T, F_R2) => I_O2
- ....
- And(F_T, F_R[L]) => I_O[L]

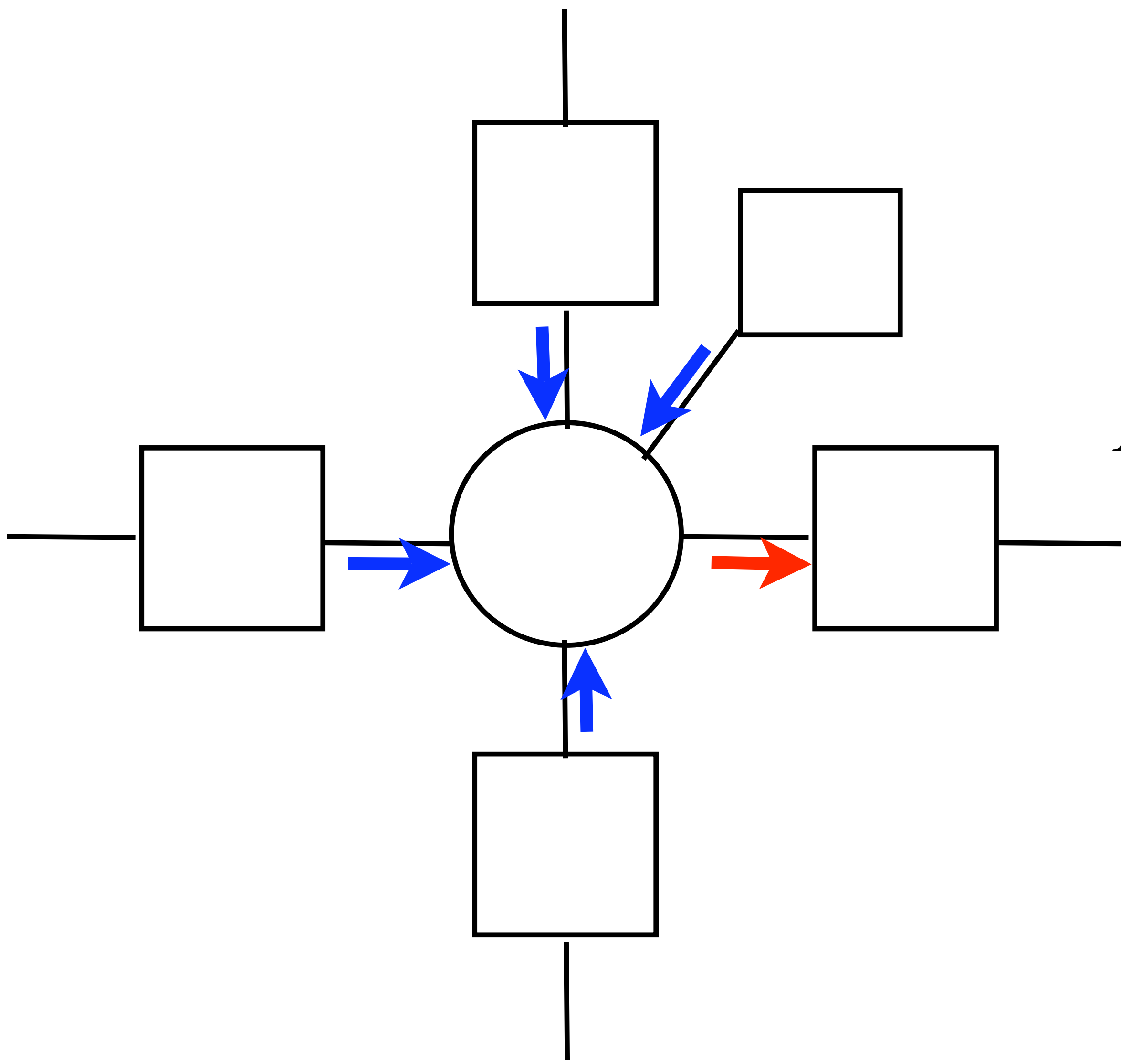
Smoothing by Weighted-Least-Squares



$$E = \sum_i R_i (x_i - y_i)^2 + \sum_{(ij)} (x_i - x_j)^2$$

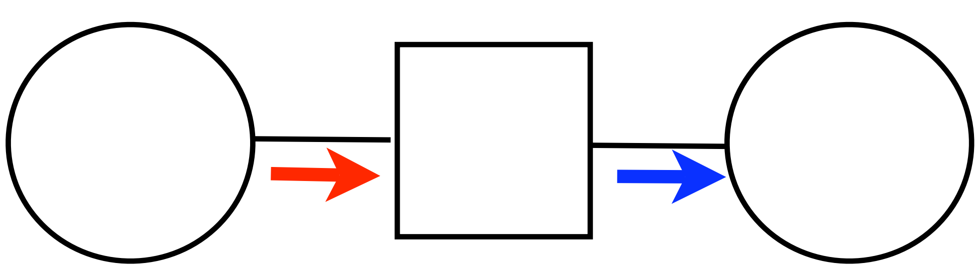
“Standard” Message-Passing Solution:

$$\hat{x}_i = \frac{\sum_{a \in N(i)} R_{a \to i} M_{a \to i}}{\sum_{a \in N(i)} R_{a \to i}}$$



$$M_{i \to a} = \frac{\sum_{b \in N(i) \backslash a} R_{b \to i} M_{b \to i}}{\sum_{b \in N(i) \backslash a} R_{b \to i}}$$

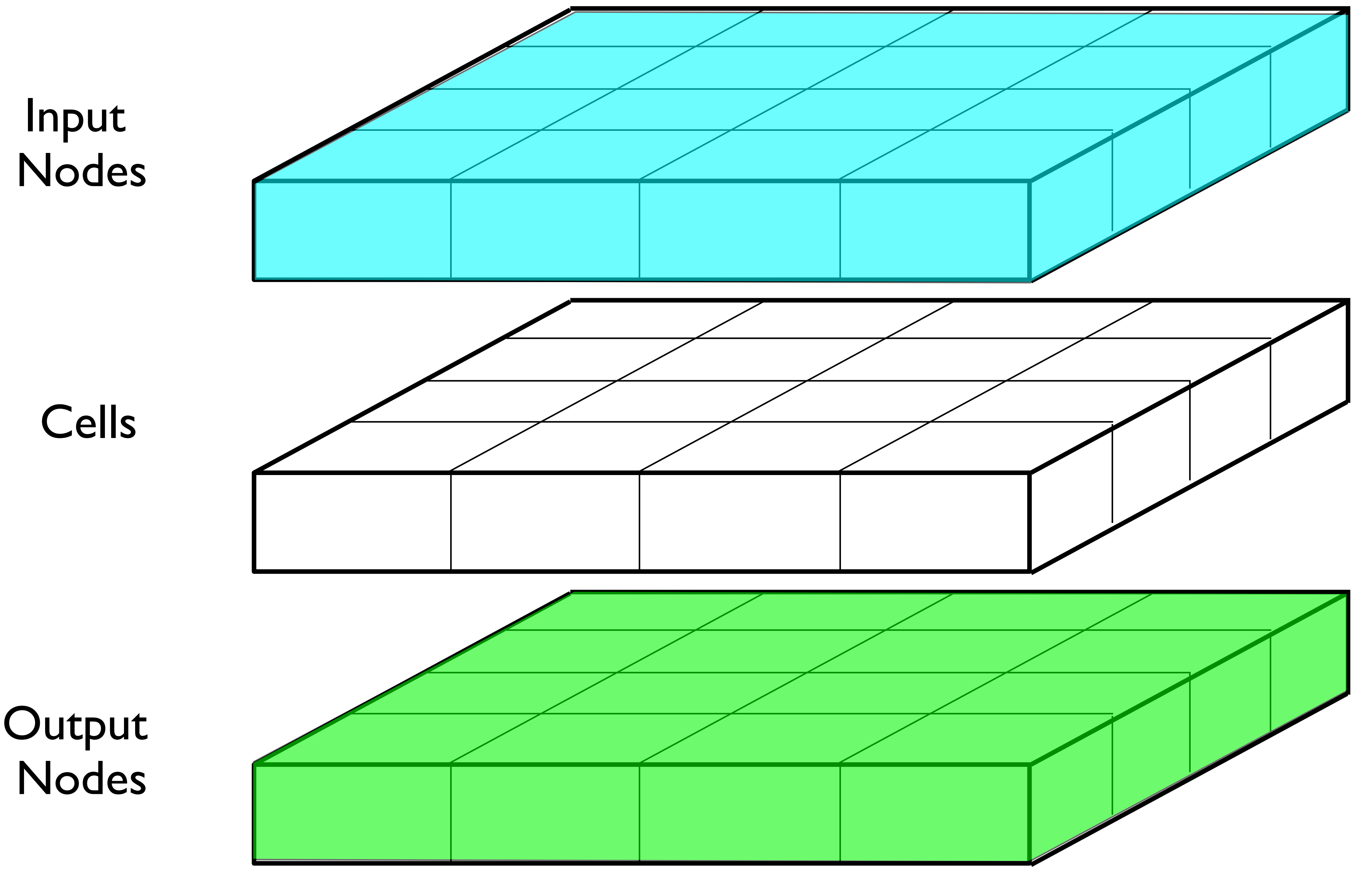
$$R_{i \to a} = \sum_{b \in N(i) \backslash a} R_{b \to i}$$



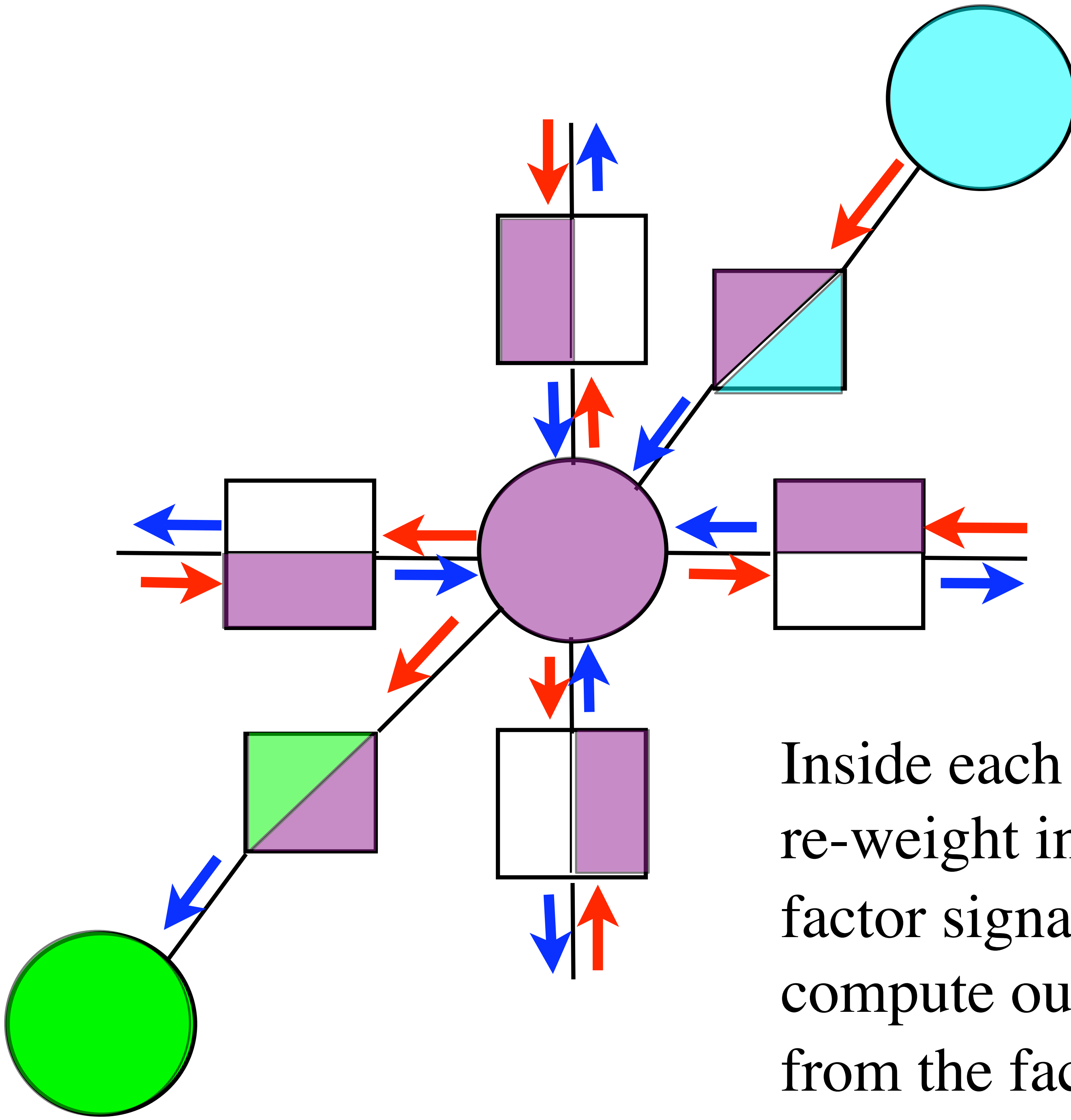
$$M_{a \to j} = M_{i \to a}$$

$$R_{a \to j} = R_{i \to a} / (1 + R_{i \to a})$$

“Multi-cellular” Solution



Dividing computational responsibilities between cells



We need 9 inter-cellular signals: one is the signal sent to the output node, plus 4 pairs of signals for the 4 different directions.

We use “receptors” that only allow inter-cellular signals from neighbors in the appropriate direction.

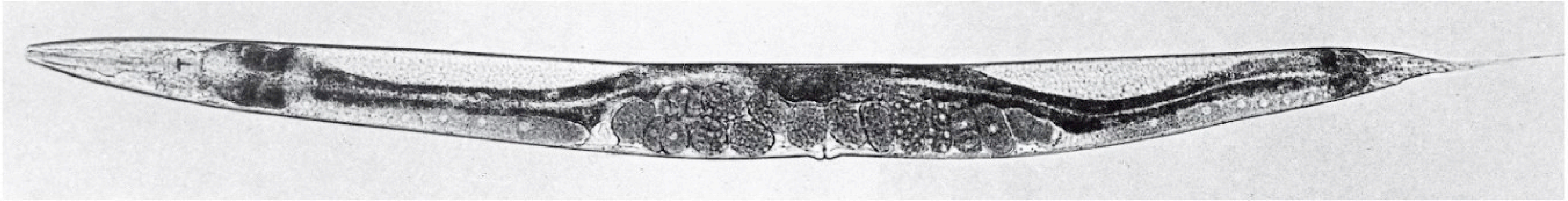
Inside each cell, we include logic units that re-weight incoming signals into internal factor signals, as well as logic units that compute outgoing intercellular signals from the factor signals.

Advantages of multi-cellular logic circuits

- Can compactly specify a very large network
- Automatic code re-use
- Potentially convenient for mass-production
- Perhaps a compact specification is nice for genetic algorithms, or other blind search strategies.

- Using many cells with identical specifications seems to be a key part of nature's design strategy for building complex machines.

C. elegans: a favorite multi-cellular "model organism"



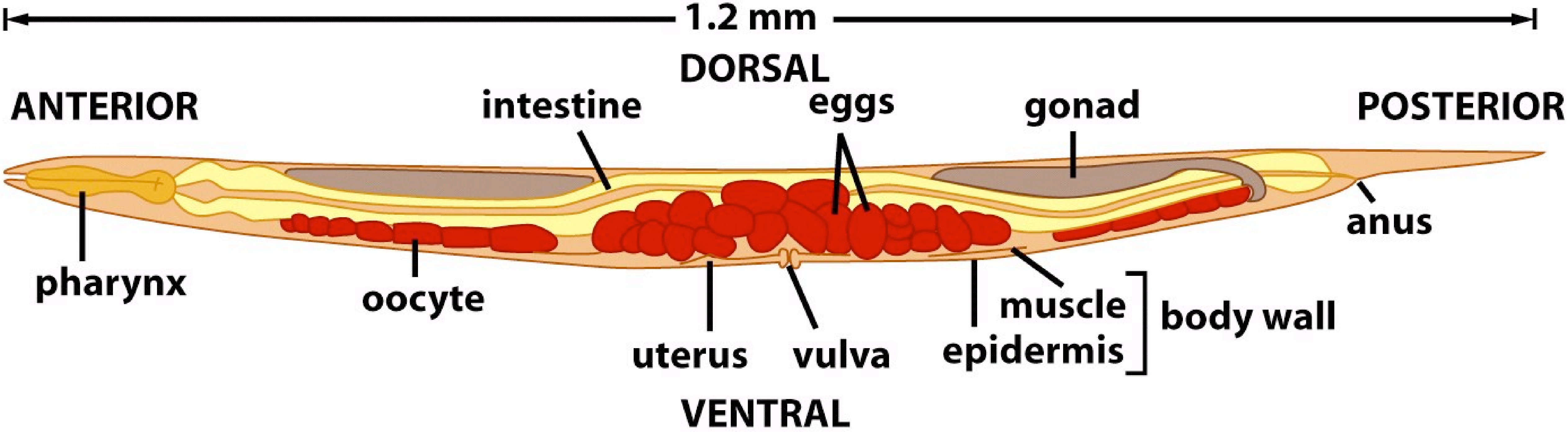
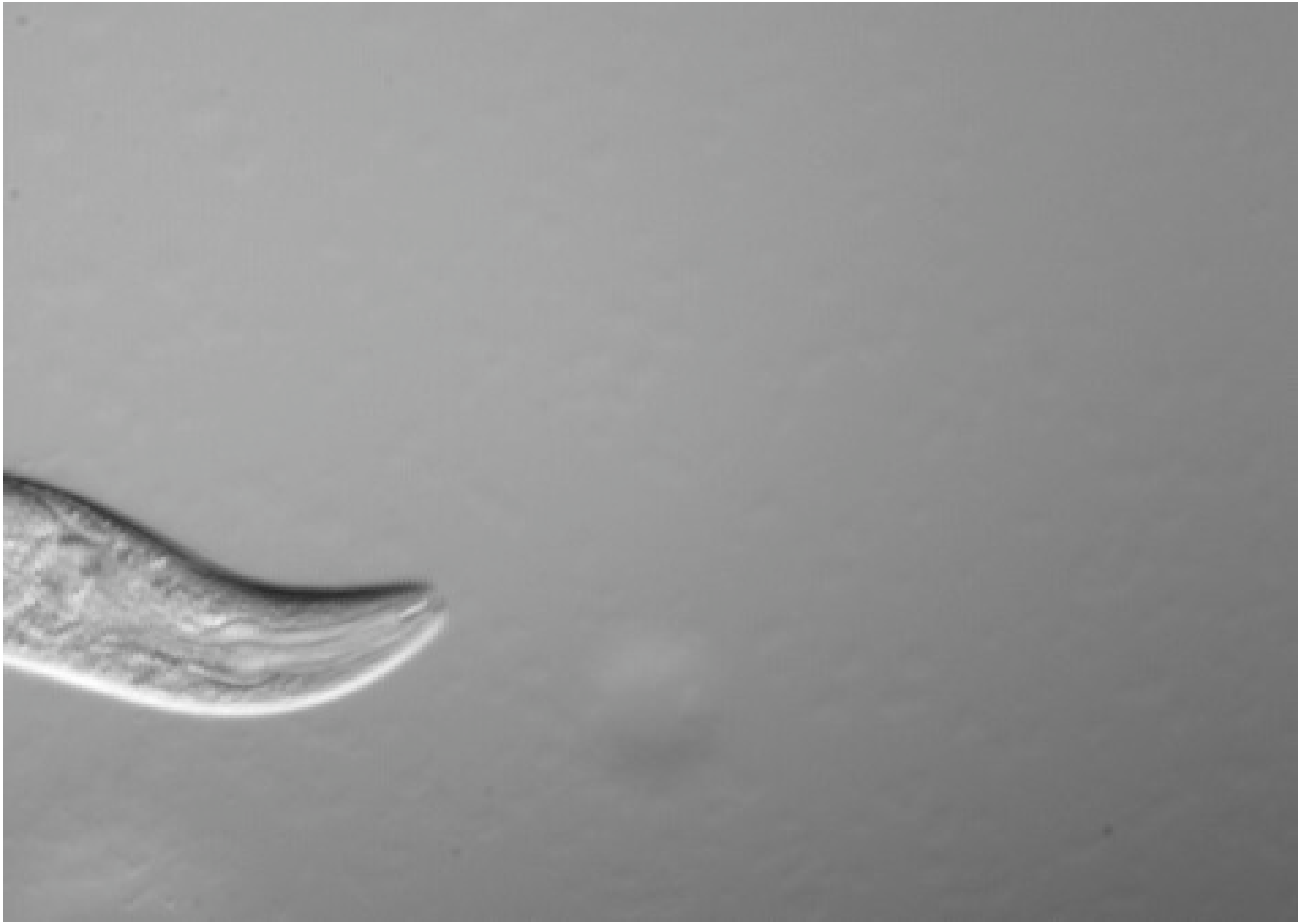
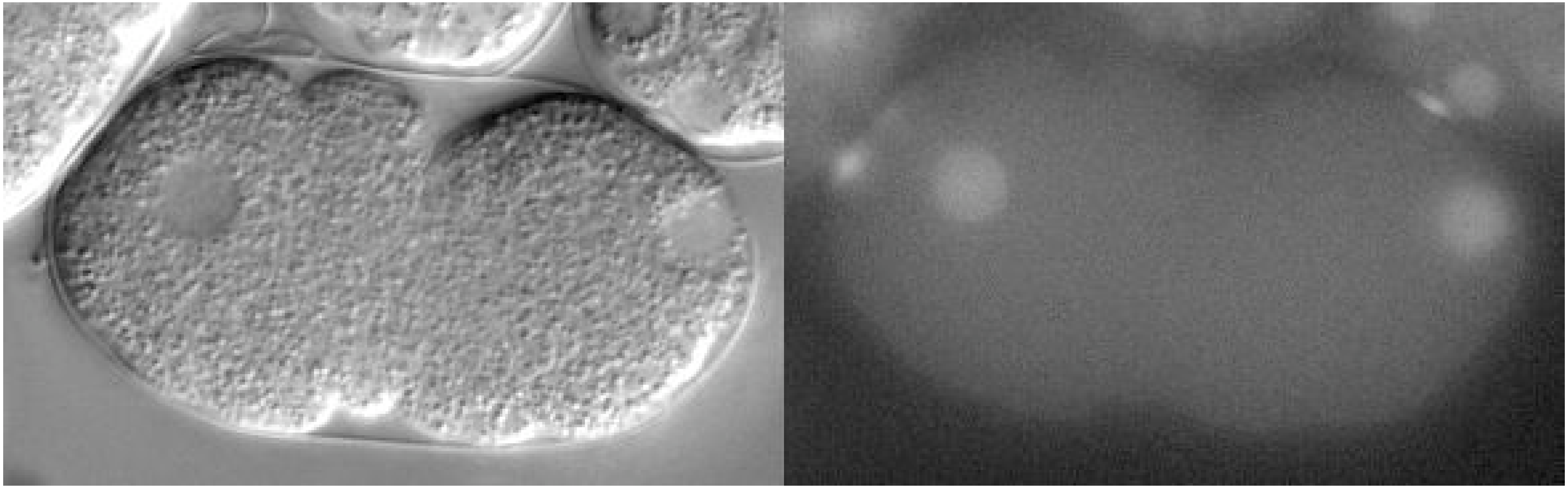


Figure 10.1a *The Biology of Cancer* (© Garland Science 2007), R. Weinberg



from B. Goldstein lab, U.N.C.

First Stages of Embryonic Development

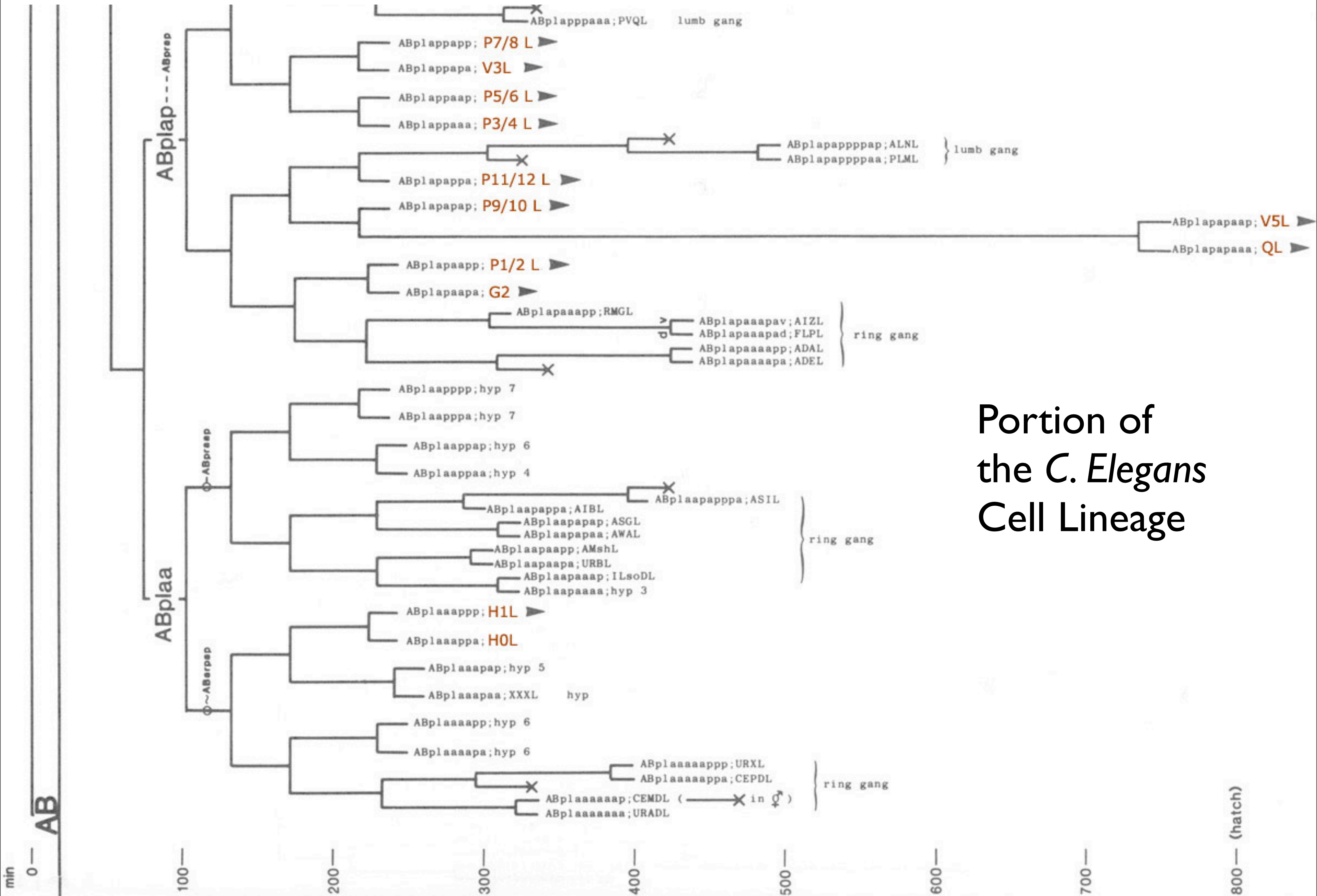


Hannak, E., et.al., J. Cell Biology, 2001

Complete Embryonic Development

from B. Goldstein lab, U.N.C.

Portion of the *C. Elegans* Cell Lineage



Intra-cellular Signals

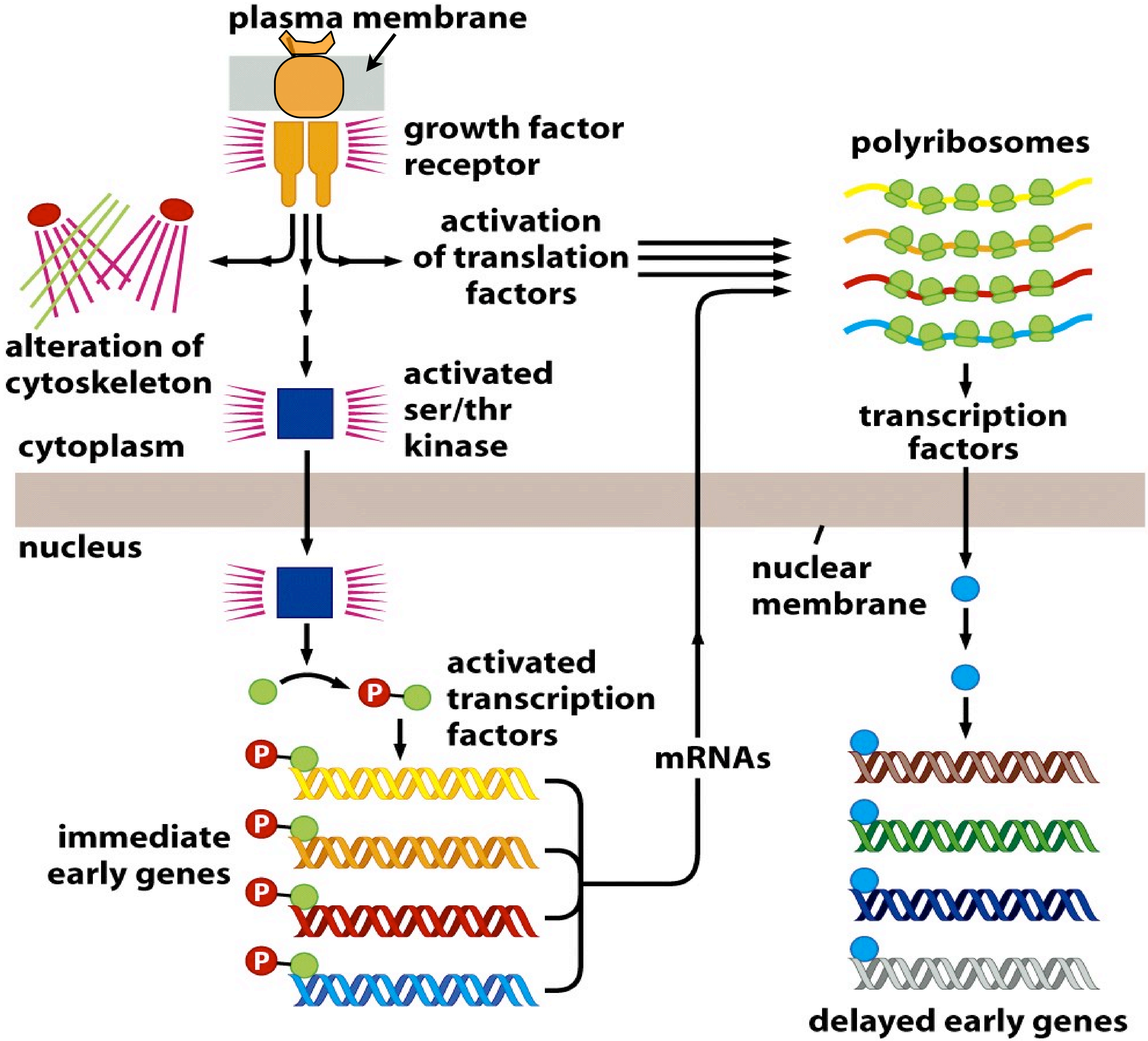


Figure 6.3 *The Biology of Cancer* (© Garland Science 2007) R. Weinberg

Transcription Control

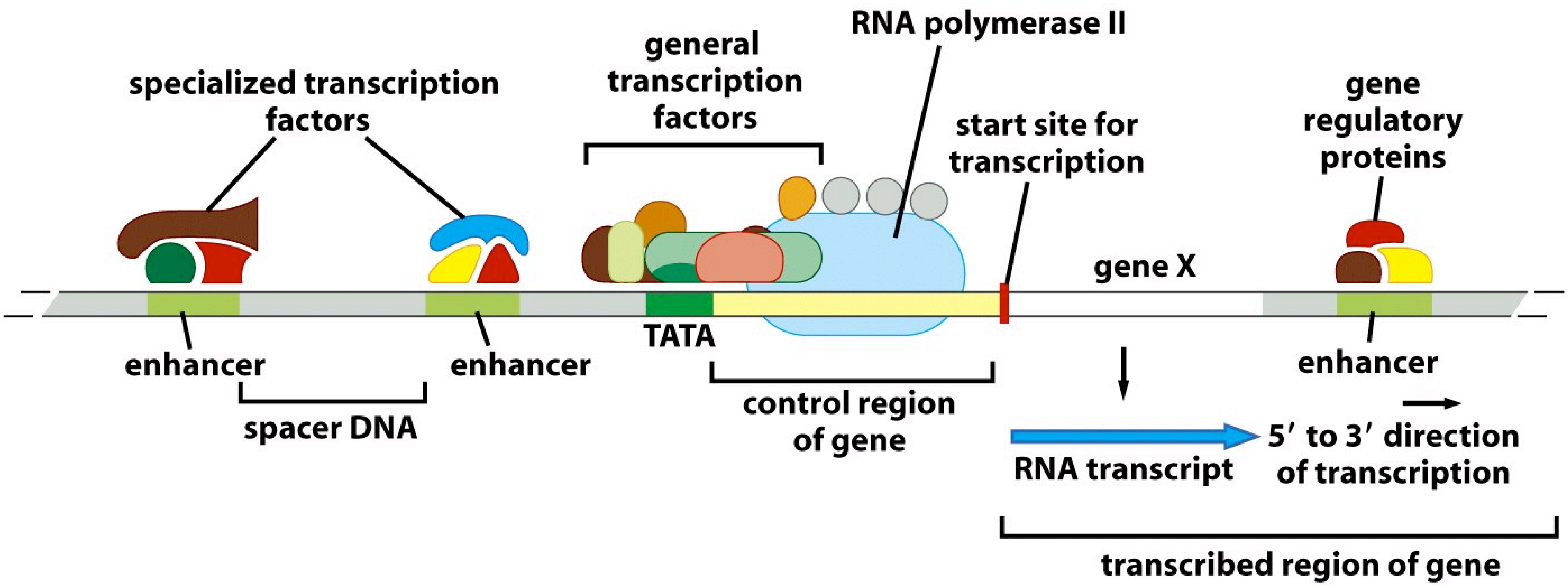


Figure 1.19 *The Biology of Cancer* (© Garland Science 2007) R. Weinberg

Biological Signal Processing

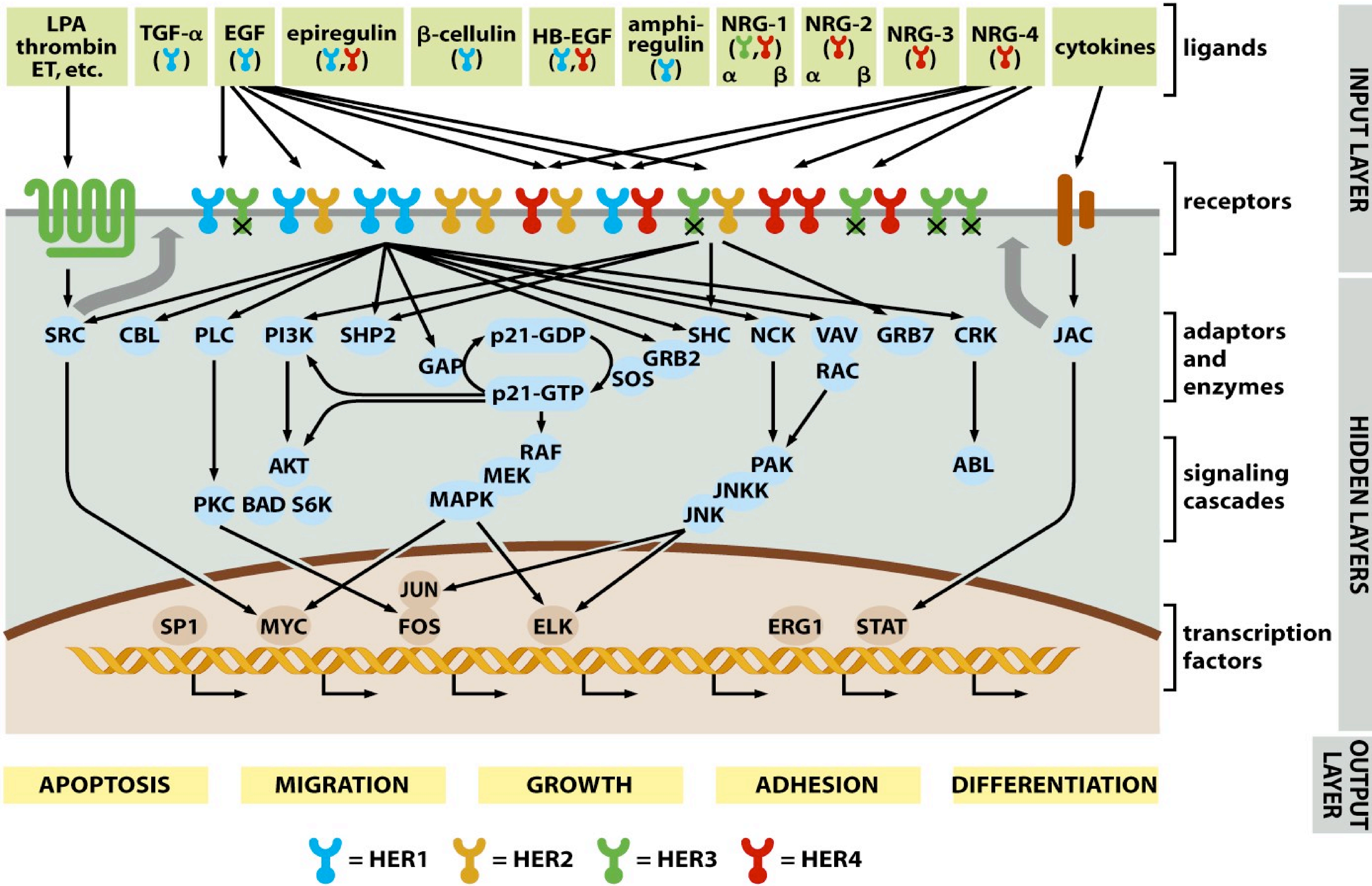
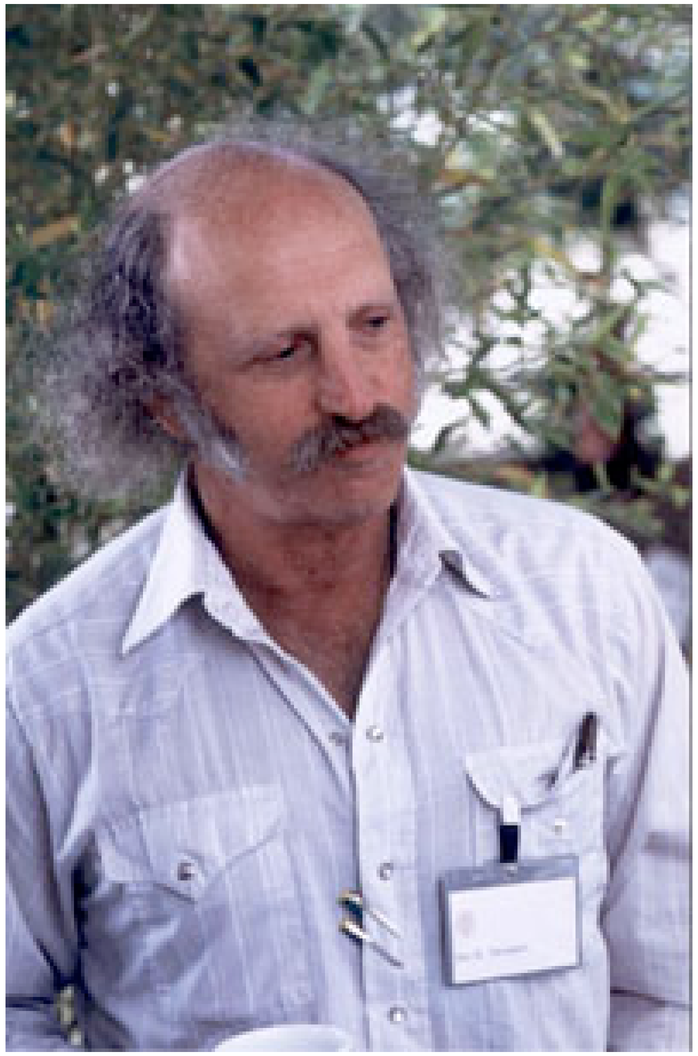
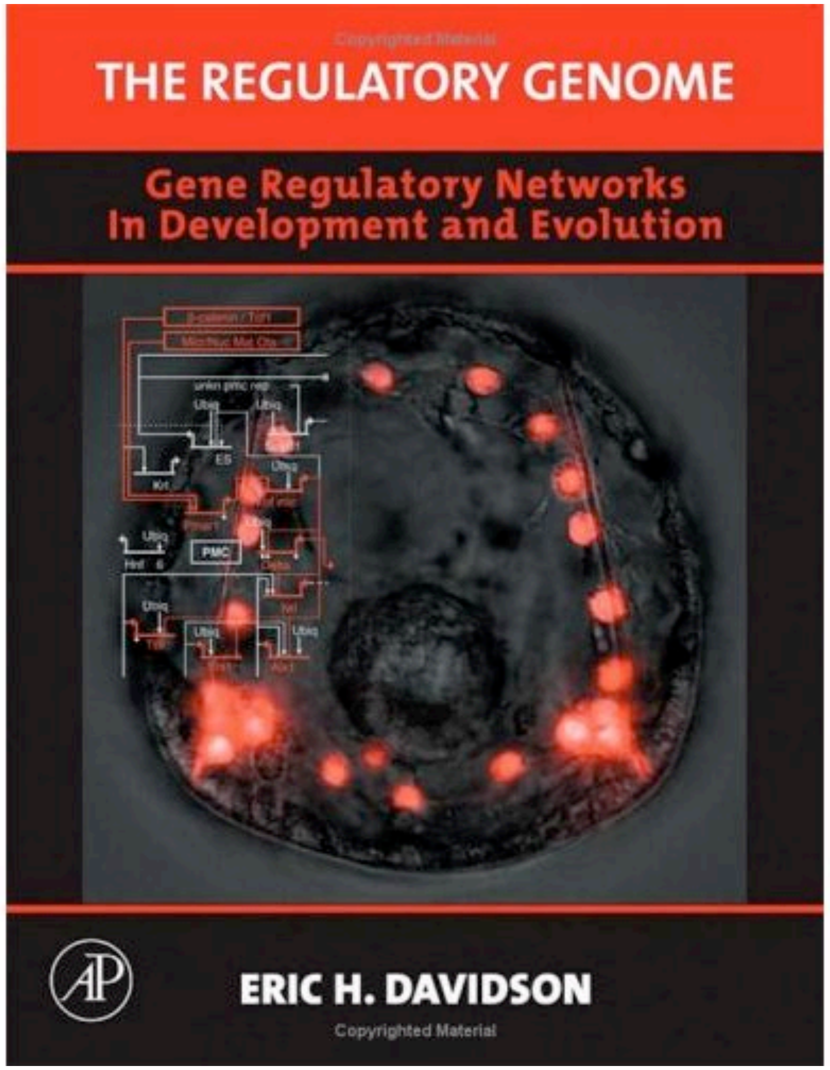


Figure 5.1 *The Biology of Cancer* (© Garland Science 2007) R. Weinberg

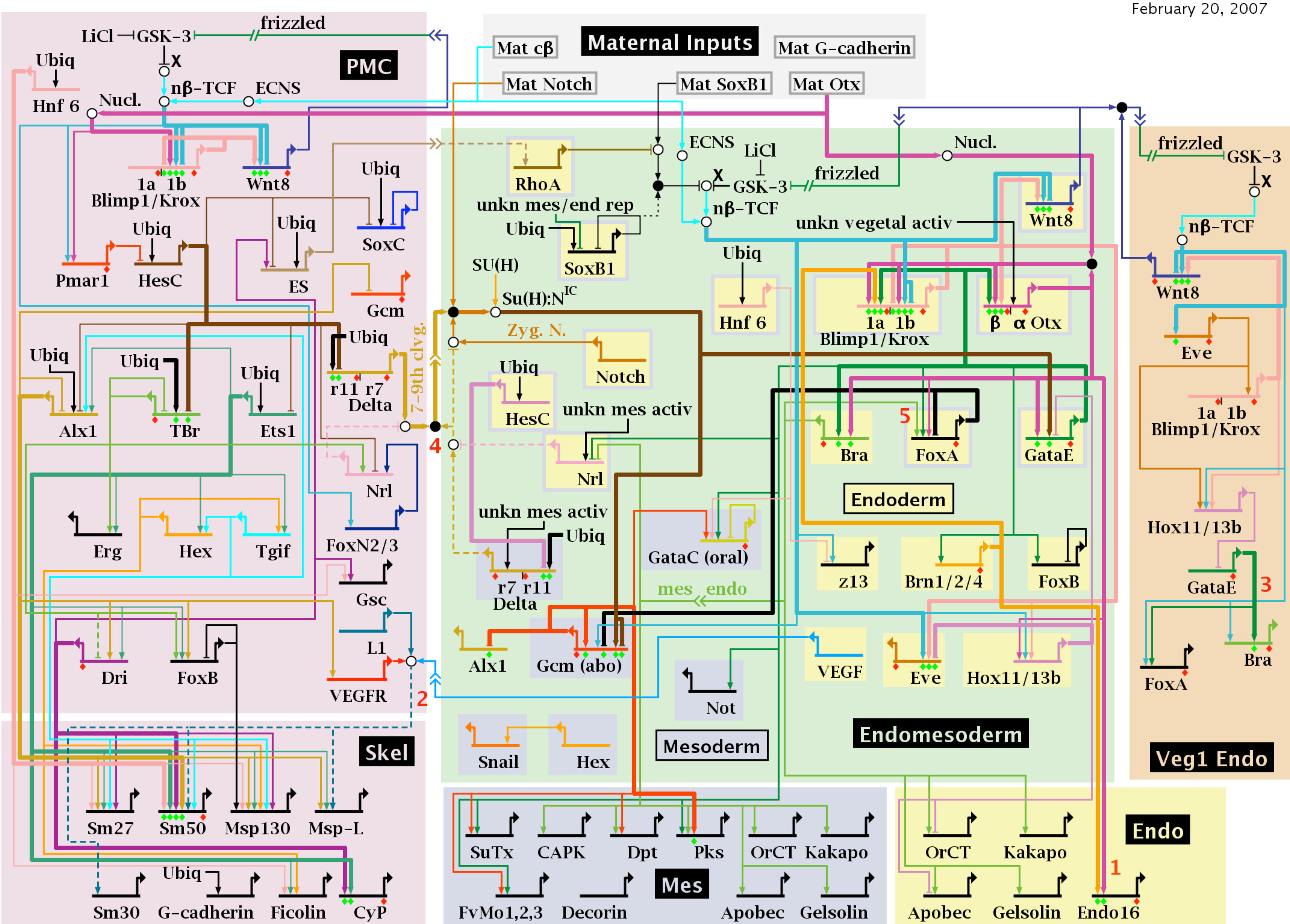

Copyright © Charles Hollahan 2006




THE REGULATORY GENOME
Gene Regulatory Networks
In Development and Evolution
ERIC H. DAVIDSON
Copyrighted Material

Endomesoderm Specification to 30 Hours

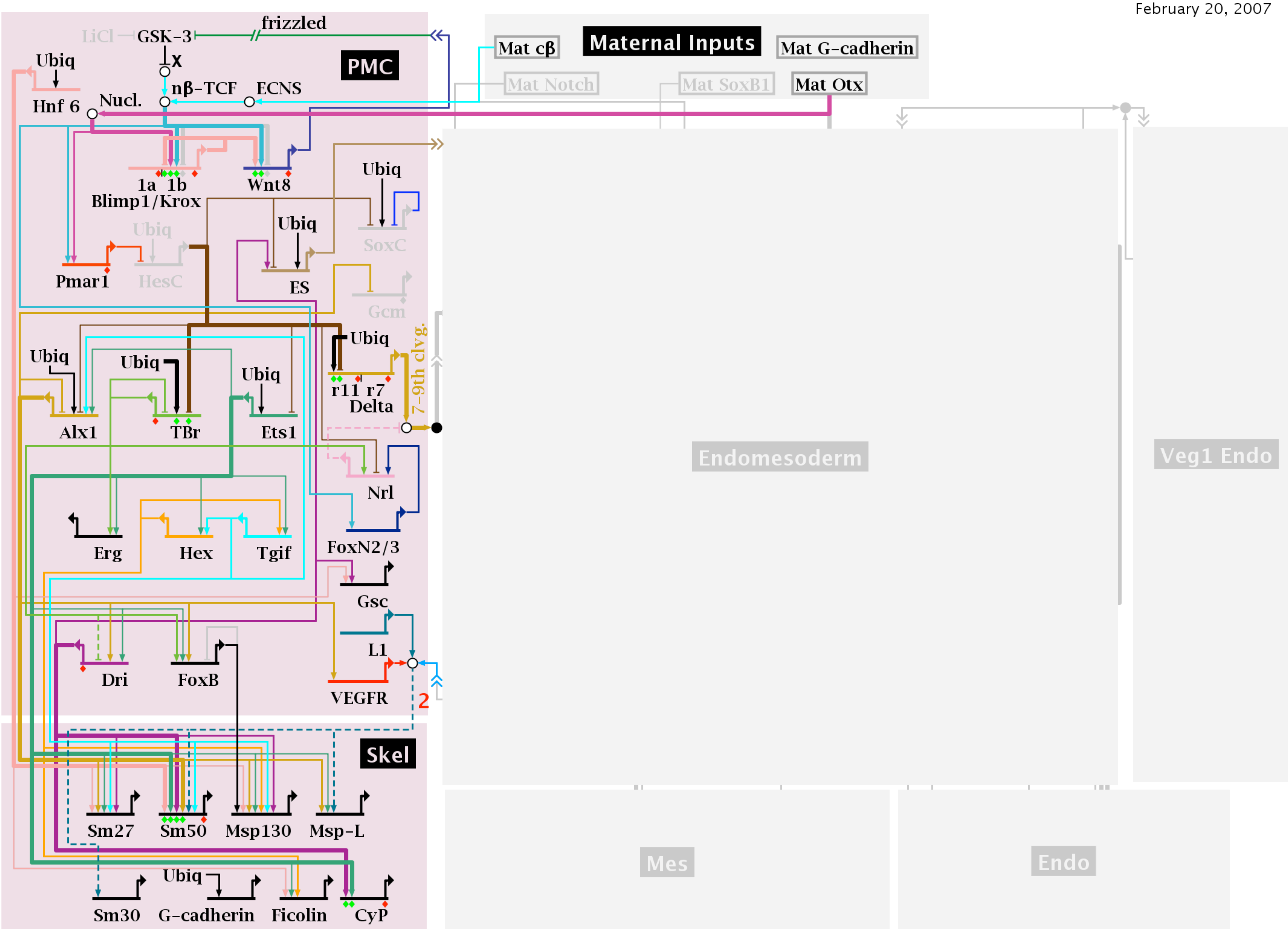
February 20, 2007



Ubiq=ubiquitous; Mat = maternal; activ = activator; rep = repressor;
unkn = unknown; Nucl. = nuclearization; χ = β-catenin source;
nβ-TCF = nuclearized b-β-catenin-Tcf1; ES = early signal;
ECNS = early cytoplasmic nuclearization system; Zyg. N. = zygotic Notch

PMC 6-30 Hours

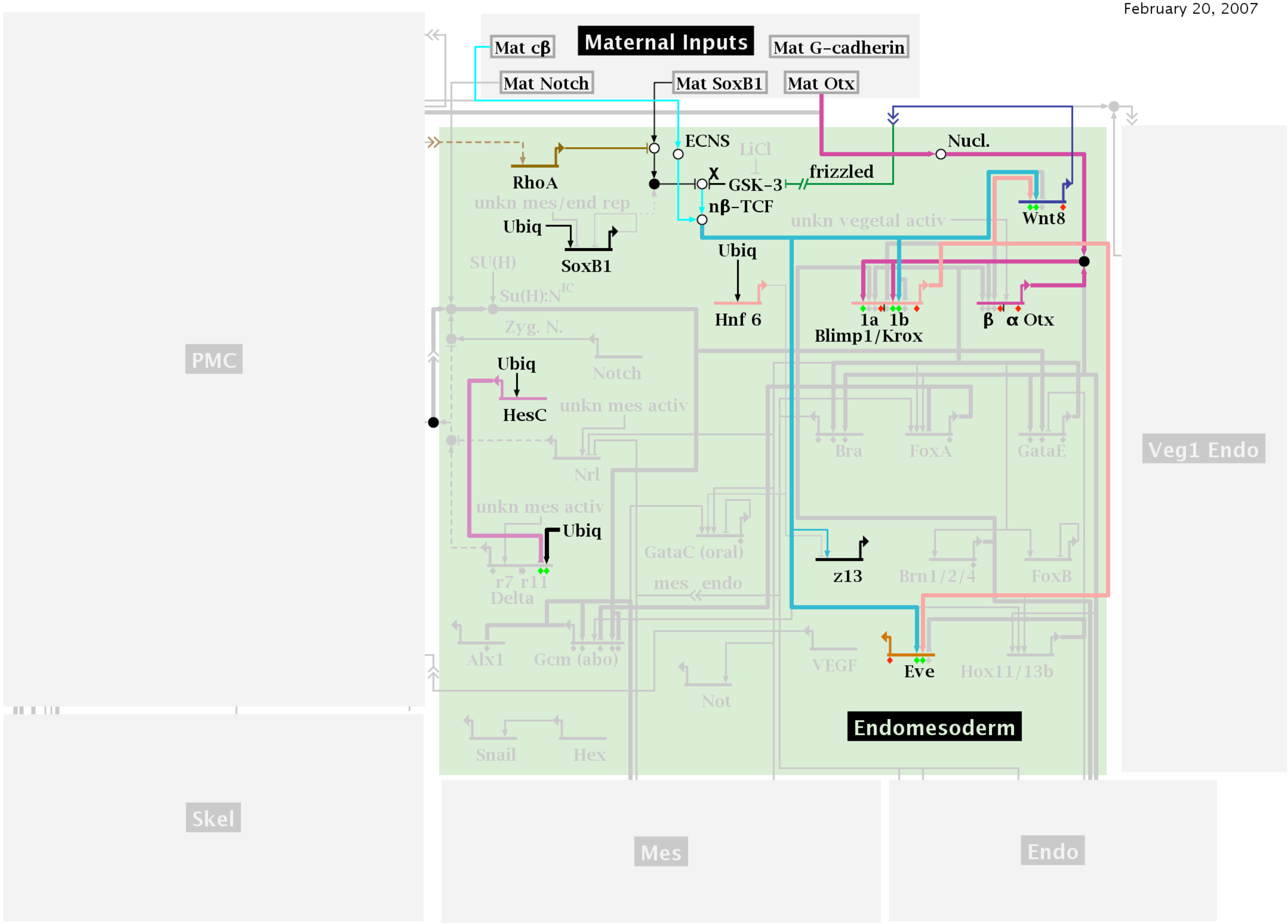
February 20, 2007



Ubiq=ubiquitous; Mat = maternal; activ = activator; rep = repressor;
unkn = unknown; Nucl. = nuclearization; χ = β-catenin source;
nβ-TCF = nuclearized b-β-catenin-Tcf1; ES = early signal;
ECNS = early cytoplasmic nuclearization system; Zyg. N. = zygotic Notch

Endomesoderm 6-11 Hours

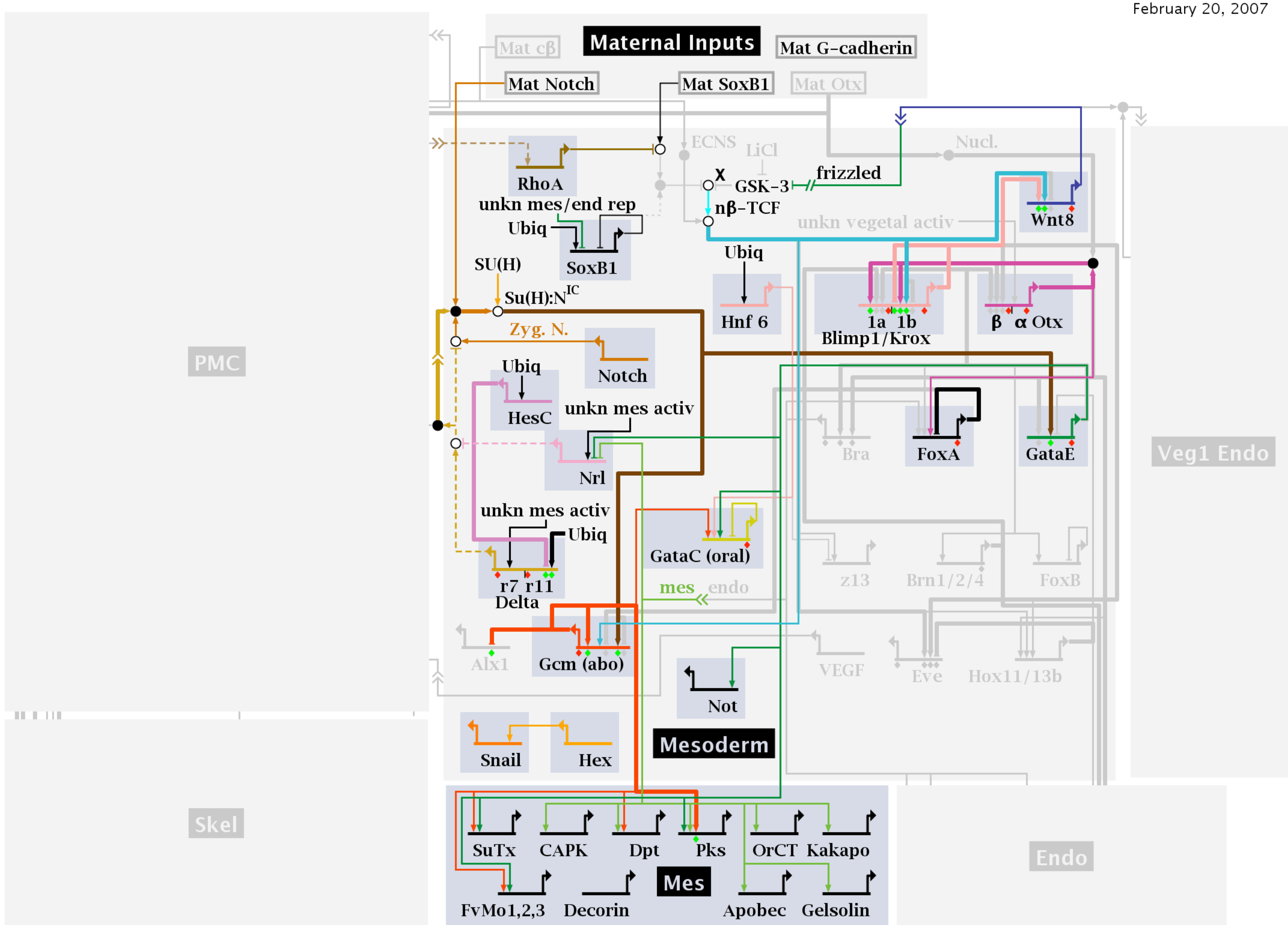
February 20, 2007



Ubiq=ubiquitous; Mat = maternal; activ = activator; rep = repressor;
unkn = unknown; Nucl. = nuclearization; χ = β-catenin source;
nβ-TCF = nuclearized b-β-catenin-Tcf1; ES = early signal;
ECNS = early cytoplasmic nuclearization system; Zyg. N. = zygotic Notch

Mesoderm 12-30 Hours

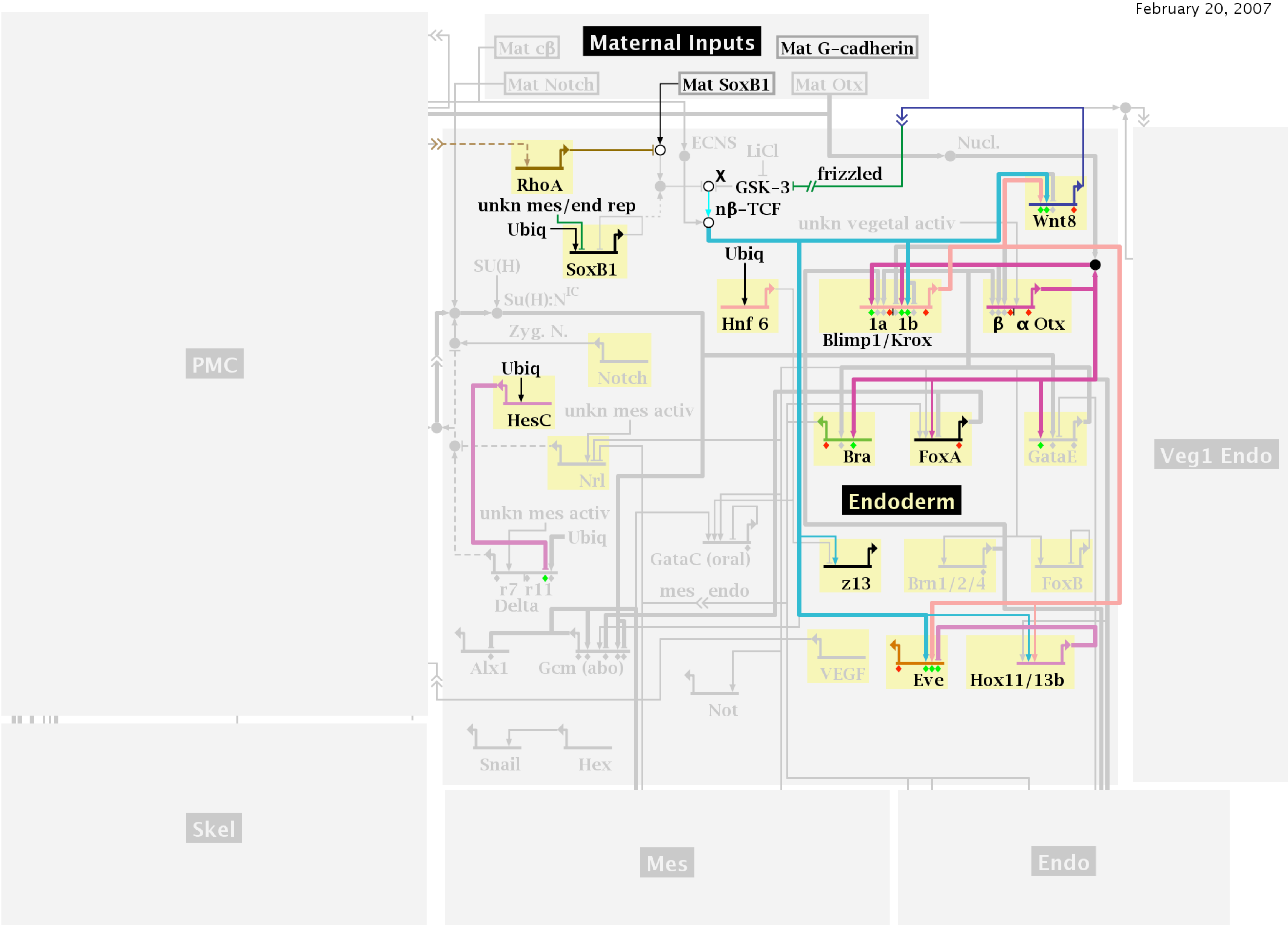
February 20, 2007



Ubiq=ubiquitous; Mat = maternal; activ = activator; rep = repressor;
unkn = unknown; Nucl. = nuclearization; χ = β-catenin source;
nβ-TCF = nuclearized b-β-catenin-Tcf1; ES = early signal;
ECNS = early cytoplasmic nuclearization system; Zyg. N. = zygotic Notch

Early Endoderm 12-17 Hours

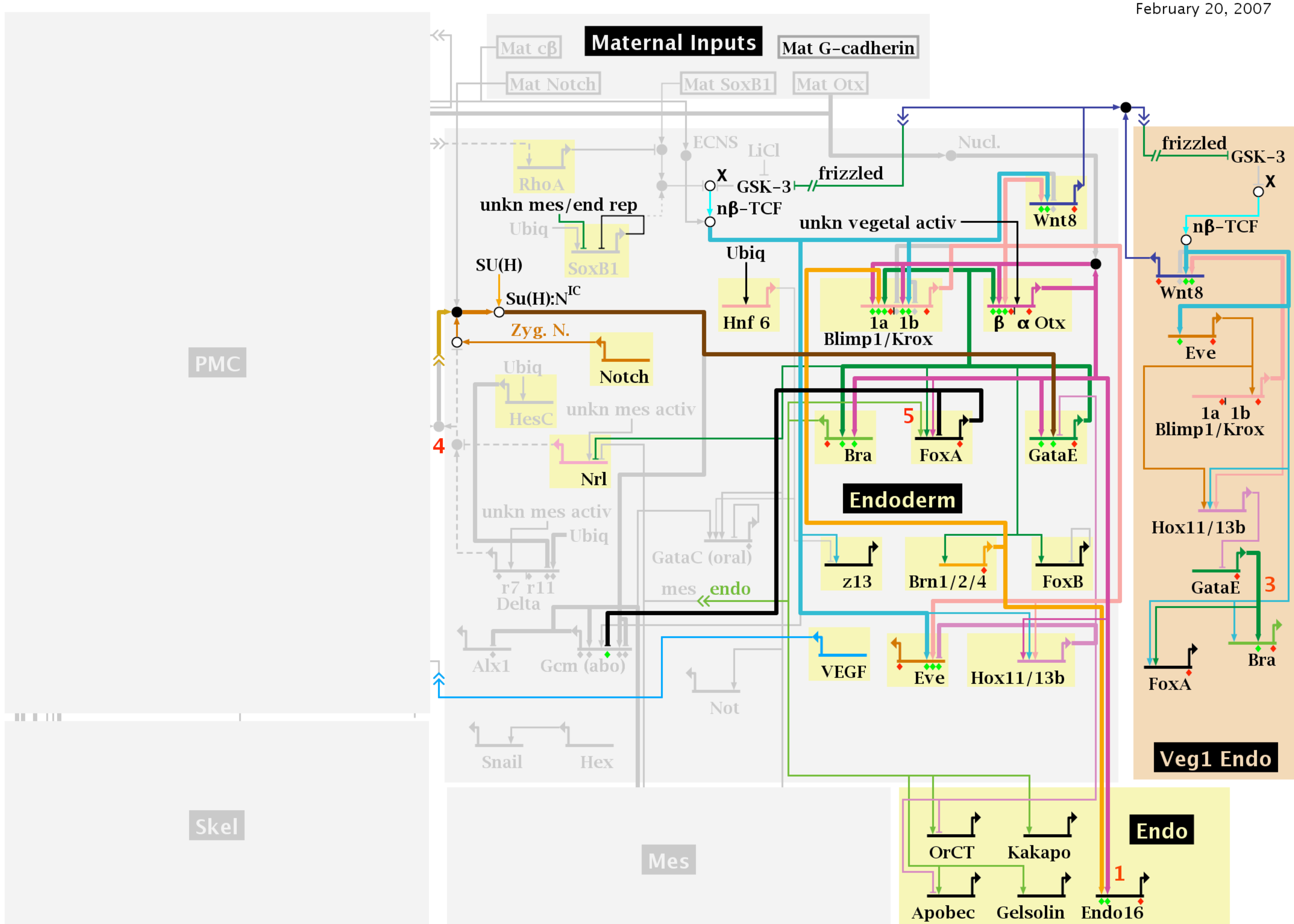
February 20, 2007



Ubiq=ubiquitous; Mat = maternal; activ = activator; rep = repressor;
unkn = unknown; Nucl. = nuclearization; χ = β-catenin source;
nβ-TCF = nuclearized b-β-catenin-Tcf1; ES = early signal;
ECNS = early cytoplasmic nuclearization system; Zyg. N. = zygotic Notch

Endoderm with Veg1 18-30 Hours

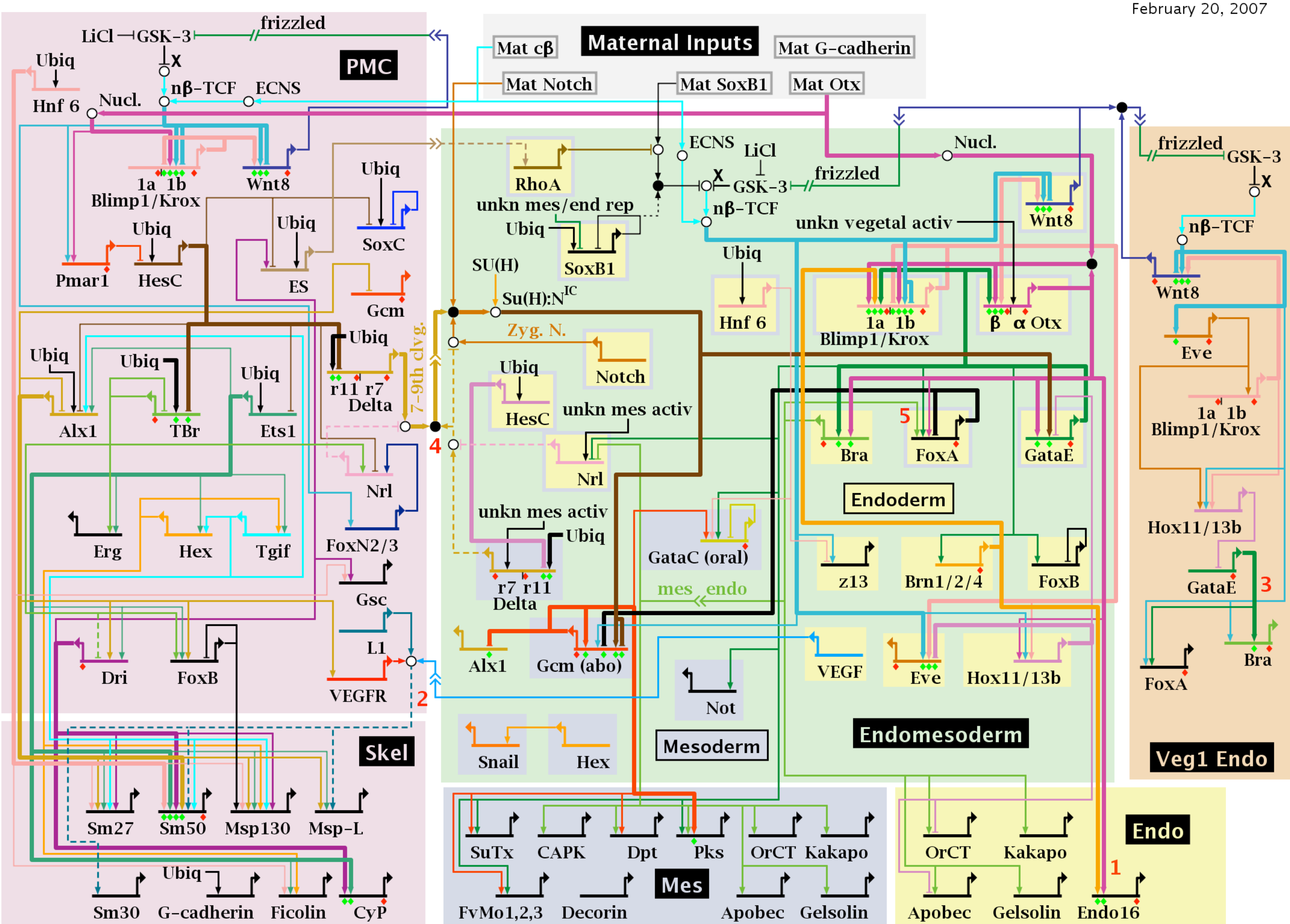
February 20, 2007



Ubiq=ubiquitous; Mat = maternal; activ = activator; rep = repressor;
unkn = unknown; Nucl. = nuclearization; χ = β-catenin source;
nβ-TCF = nuclearized b-β-catenin-Tcf1; ES = early signal;
ECNS = early cytoplasmic nuclearization system; Zyg. N. = zygotic Notch

Endomesoderm Specification to 30 Hours

February 20, 2007



Ubiq=ubiquitous; Mat = maternal; activ = activator; rep = repressor;
unkn = unknown; Nucl. = nuclearization; χ = β-catenin source;
nβ-TCF = nuclearized b-β-catenin-Tcf1; ES = early signal;
ECNS = early cytoplasmic nuclearization system; Zyg. N. = zygotic Notch

Longabaugh, et.al., Developmental Biology, 2005

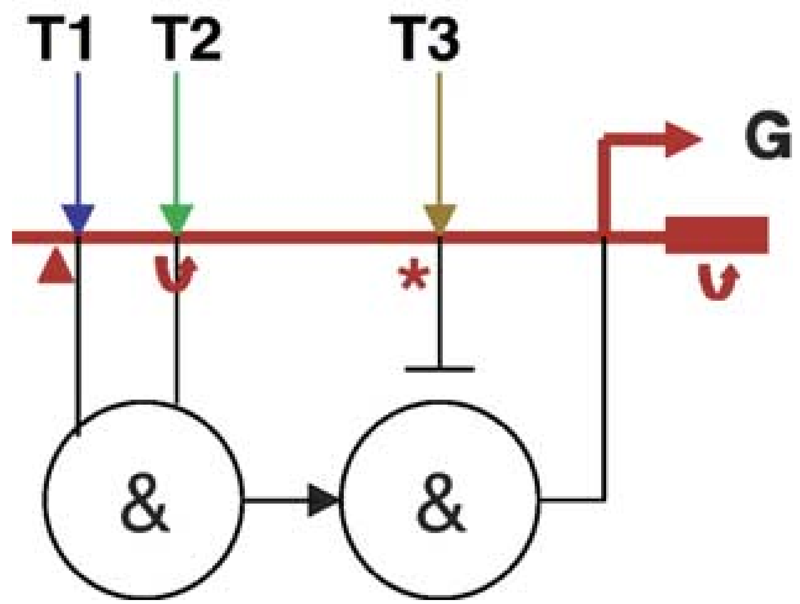


Fig. 4. Proposed DNA-based computational representation of a gene. The red horizontal line represents DNA. The portion to the left of the bent red arrow represents upstream (5′) sequence. The red box to the right of the bent red arrow represents a DNA feature, such as the first exon. T1–3 are transcription factors, which in this example bind the upstream sequence and transcriptionally regulate the expression of *G*. The regulatory interactions of the three transcription factors are represented symbolically by the two circles labeled with the logical AND symbol. The bar at the end of the line from T3 to the right hand circle indicates T3 activity acts as a repressor. Since the other input to this interaction is the logical AND of T1 and T2, the output of the second interaction (and hence gene G) can be seen to be ((T1 AND T2) AND NOT(T3)); that is, transcription of *G* is active if T1 and T2 are both active, repressed if T3 is active, and basal otherwise. The symbols just below the line representing DNA are icons for hyperlinks to genome browsers showing detailed sequence annotations such as exons (right-hand curved arrow) known transcription factor binding sites (*) and results from DNA search algorithms (triangle).

Future Directions

- ***Evolve*** multi-cellular logic circuits
- Design/Evolve more complex circuits where cells perform clearly different functions (e.g. a CPU)
- All sorts of "biological" modeling possibilities: "mating" circuits, an embryo circuit developing inside a "mother" circuit, etc.
- Multi-cellular organisms with physical structure: add motors and springs.