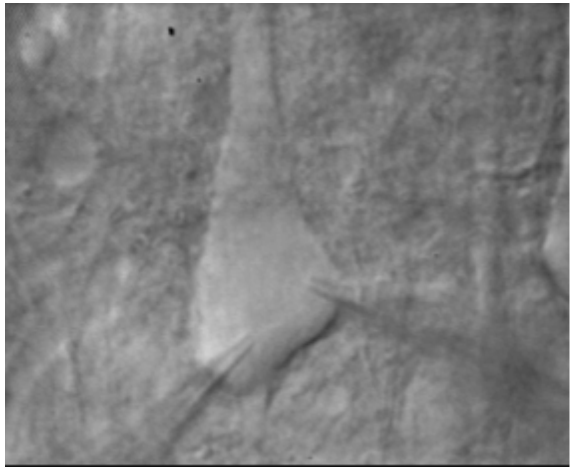
Sandia National Laboratories

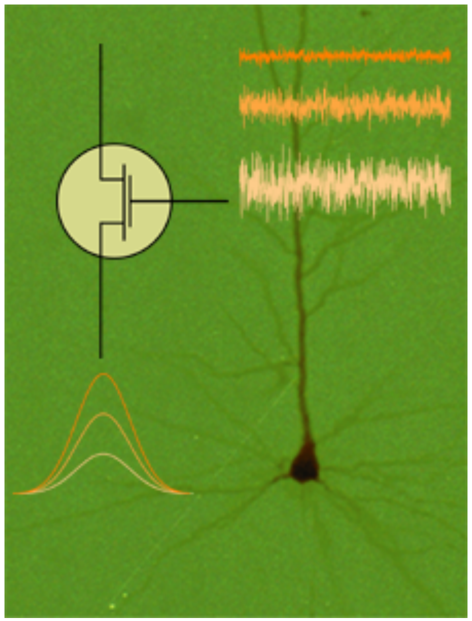
Brain-Inspired Computing at Sandia

Frances Chance
Sandia National Laboratories
March 19, 2014

Sandia National Laboratories is a multi-program laboratory managed and operated by Sandia Corporation, a wholly owned subsidiary of Lockheed Martin Corporation, for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-AC04-94AL85000. (**SAND2014-2092 C**)



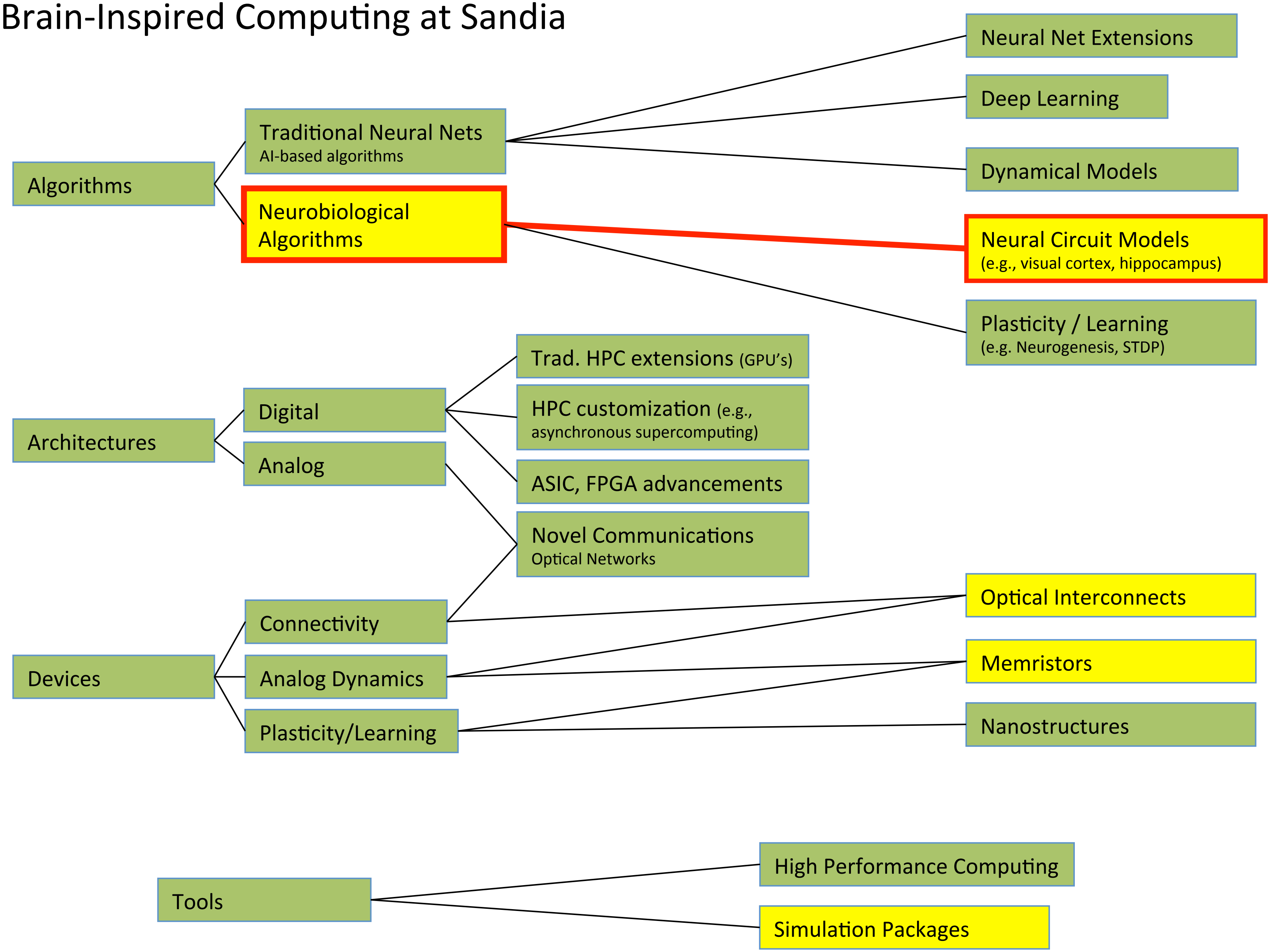
$$C\frac{dV}{dt} = g_L(E_L - V) + g_E(E_E - V)$$

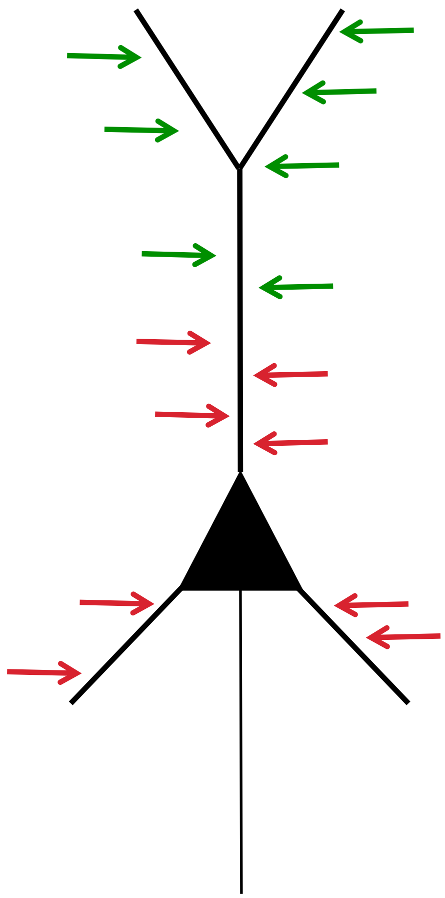


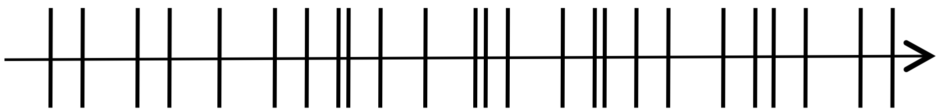
The potential of brain-inspired computing

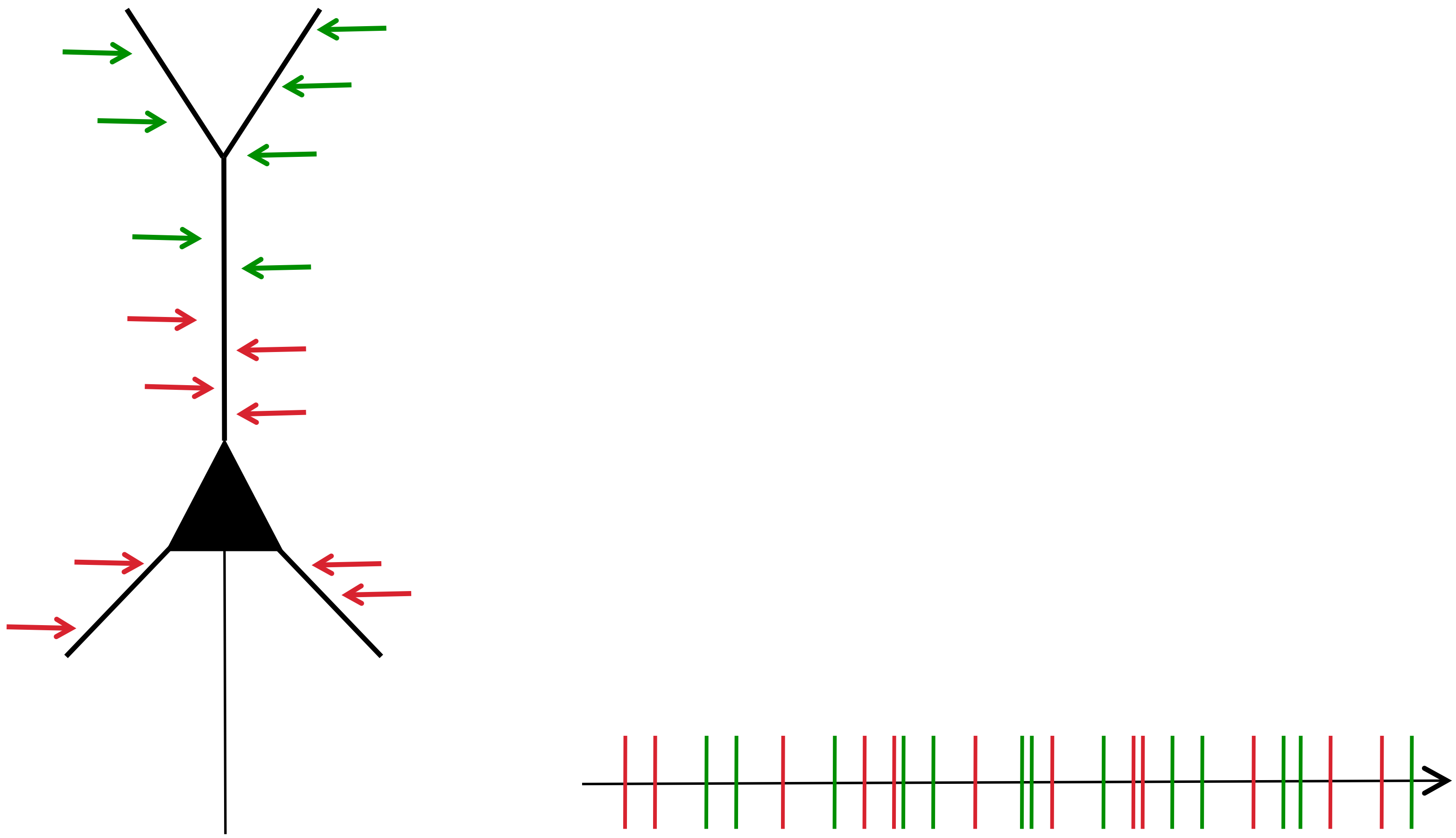
- "cognitive" tasks
  - recognition
  - inference
  - decision-making
- online learning, adaptation
- robustness
  - fault-tolerance
  - noise tolerance
- low power requirements (20W)
- relatively small size (1.3-1.4 kg)
- potential to continue to advance
  - speed of computing

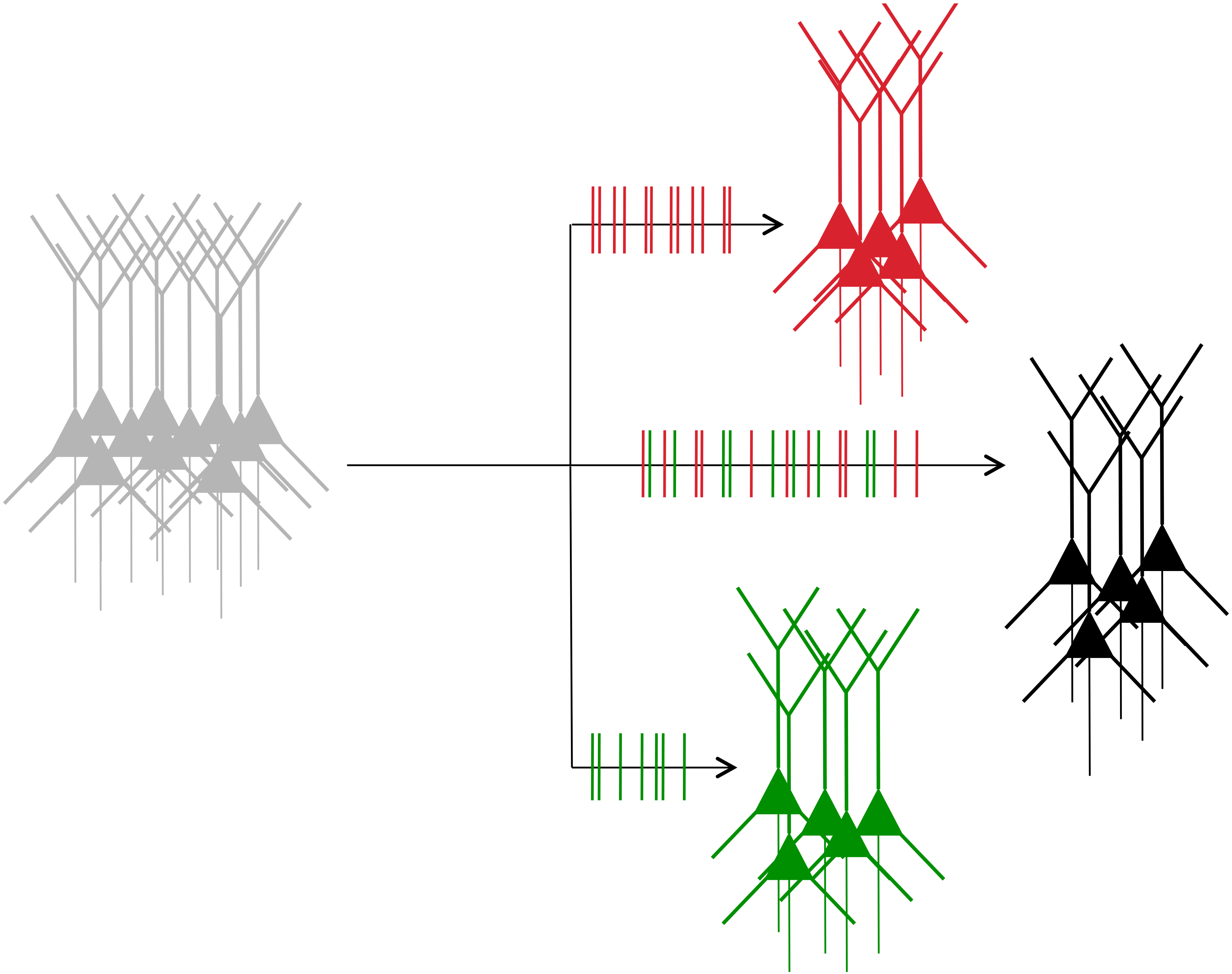
Brain-Inspired Computing at Sandia

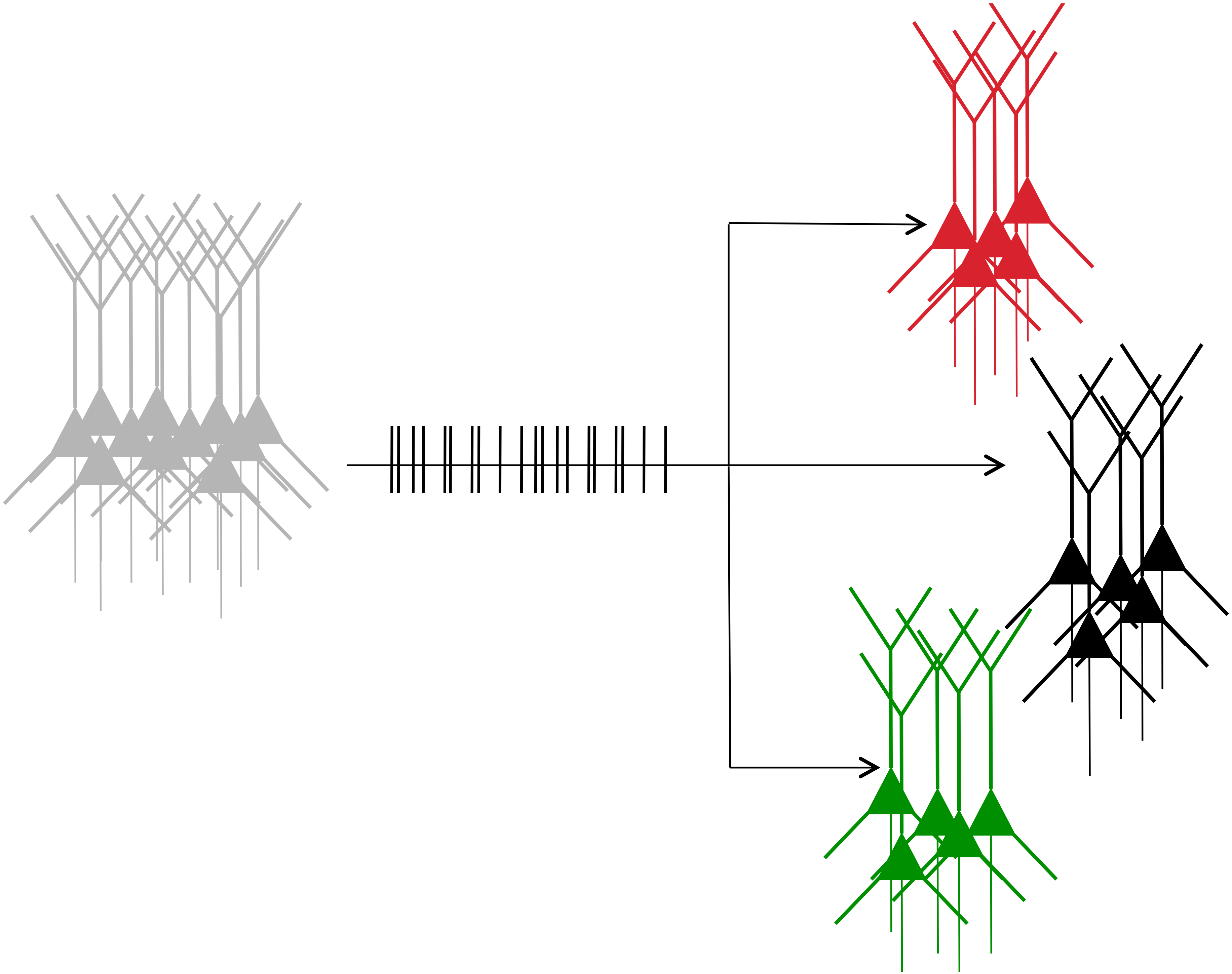


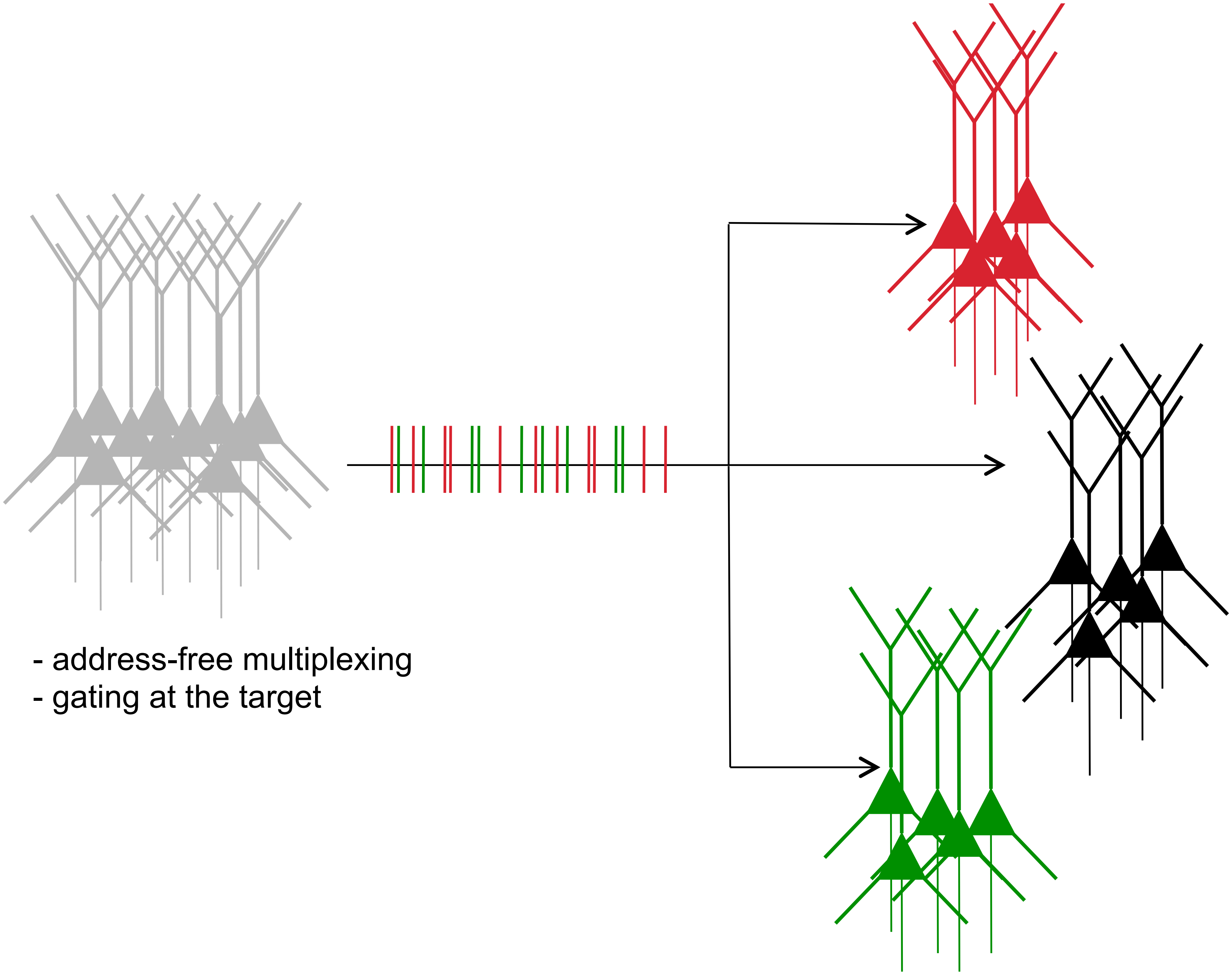




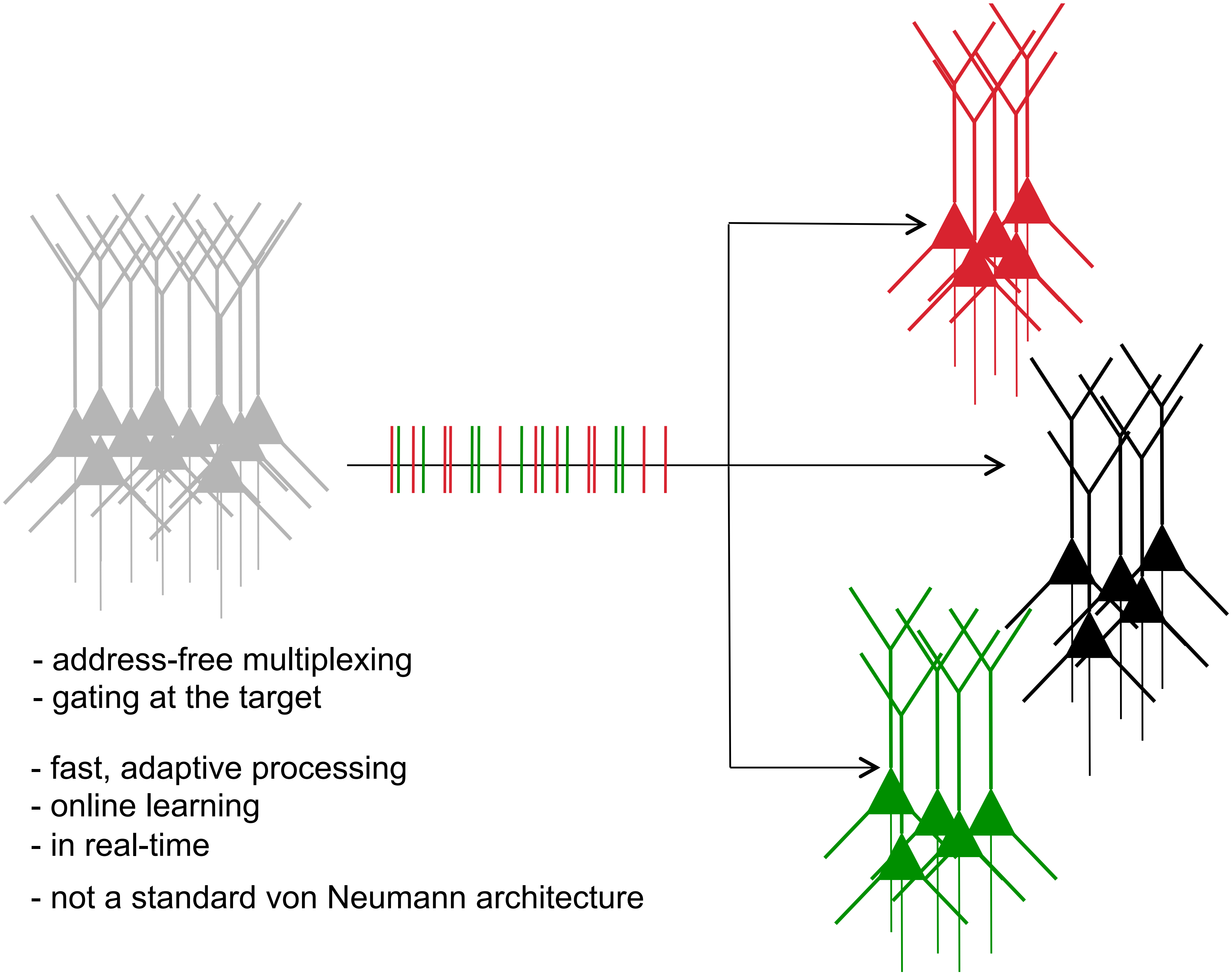








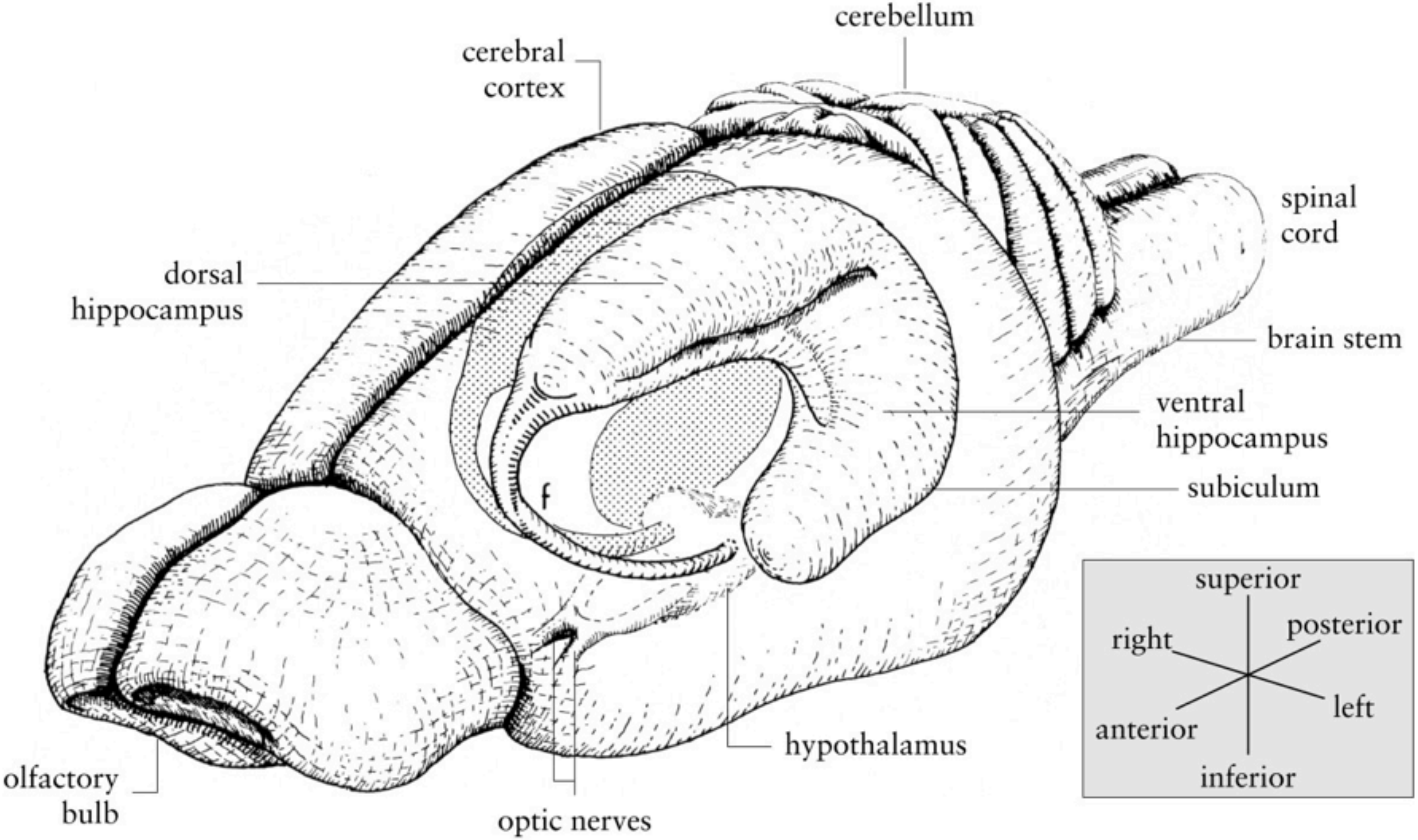
- address-free multiplexing
- gating at the target



- address-free multiplexing
- gating at the target

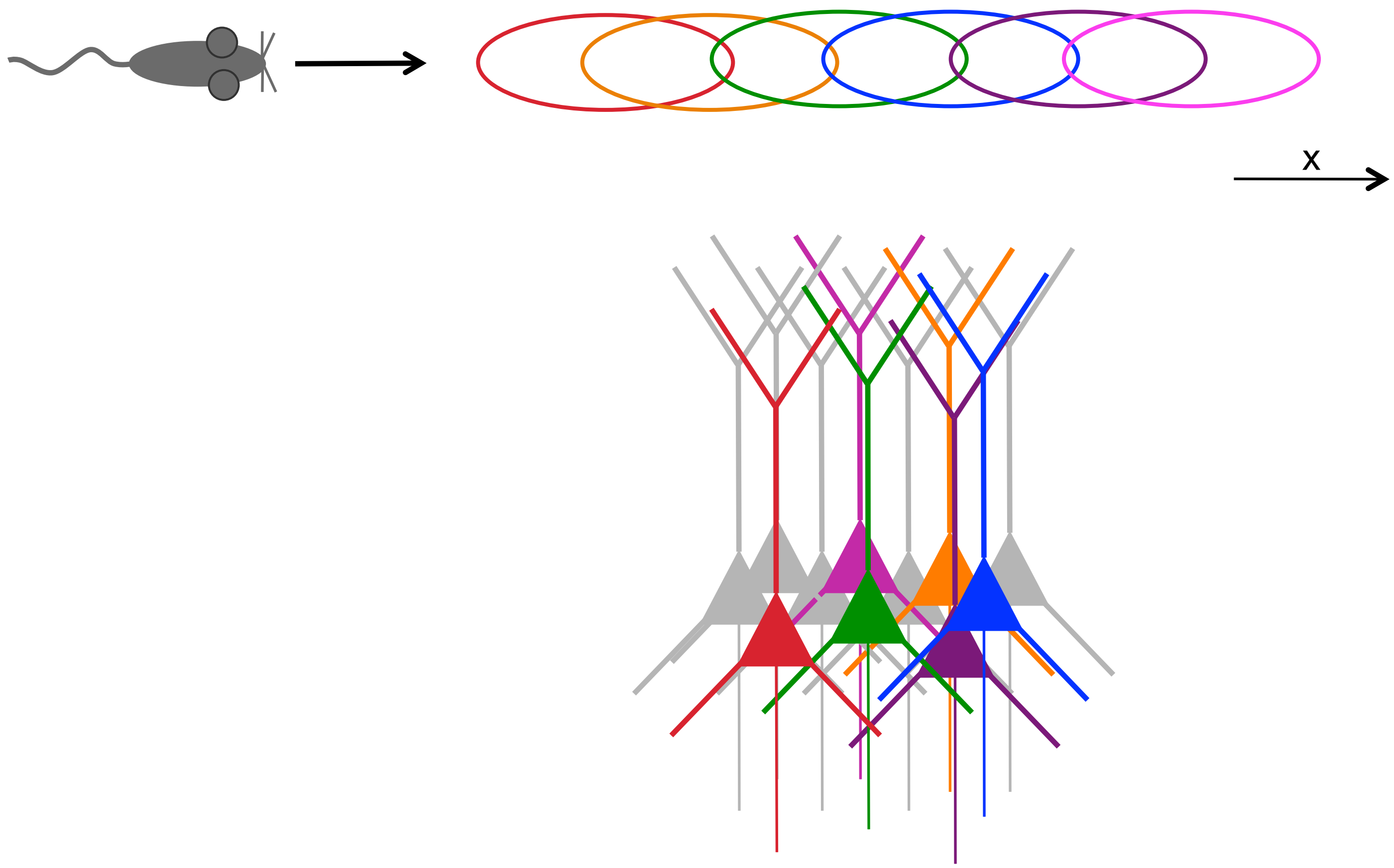
- fast, adaptive processing
- online learning
- in real-time

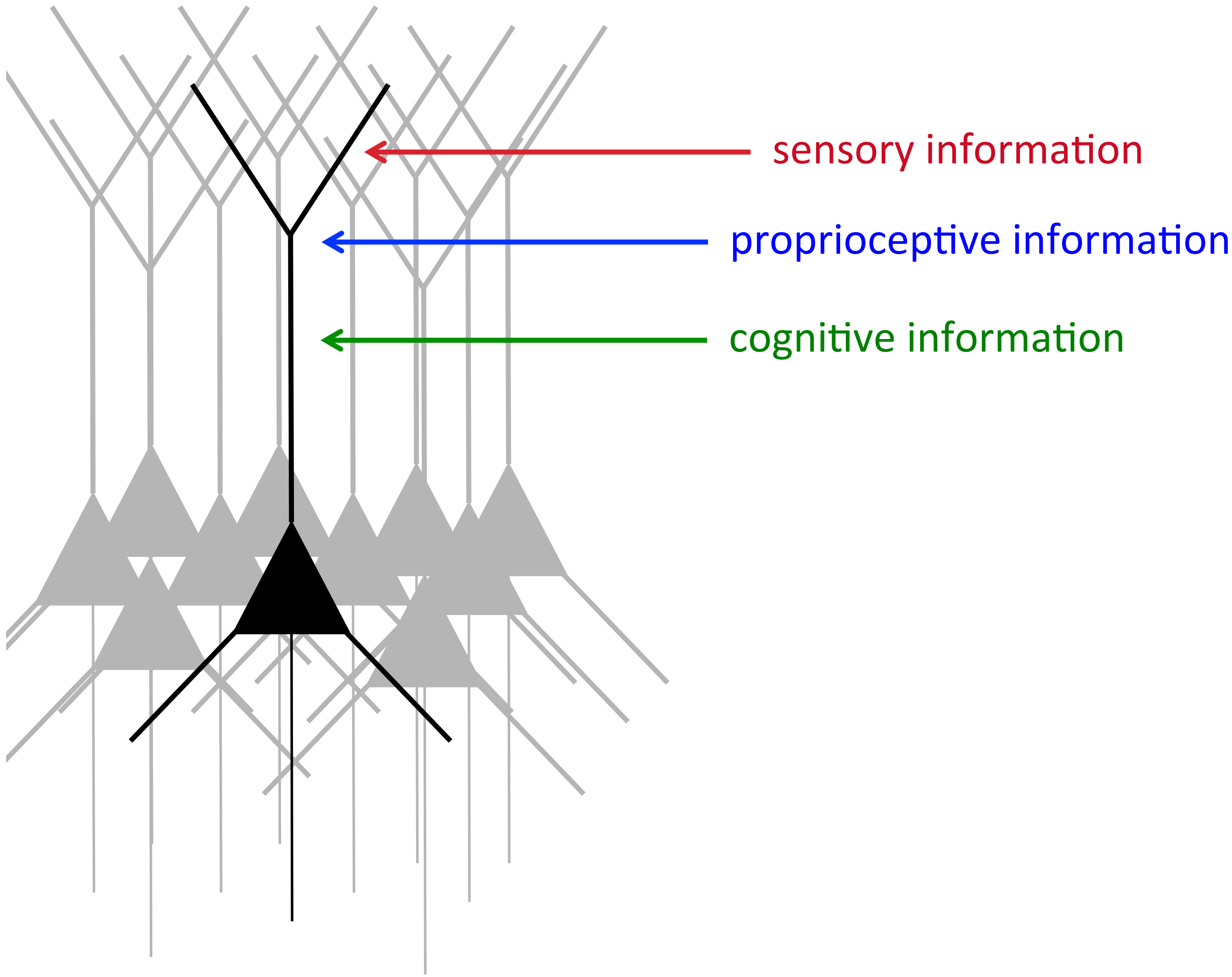
- not a standard von Neumann architecture

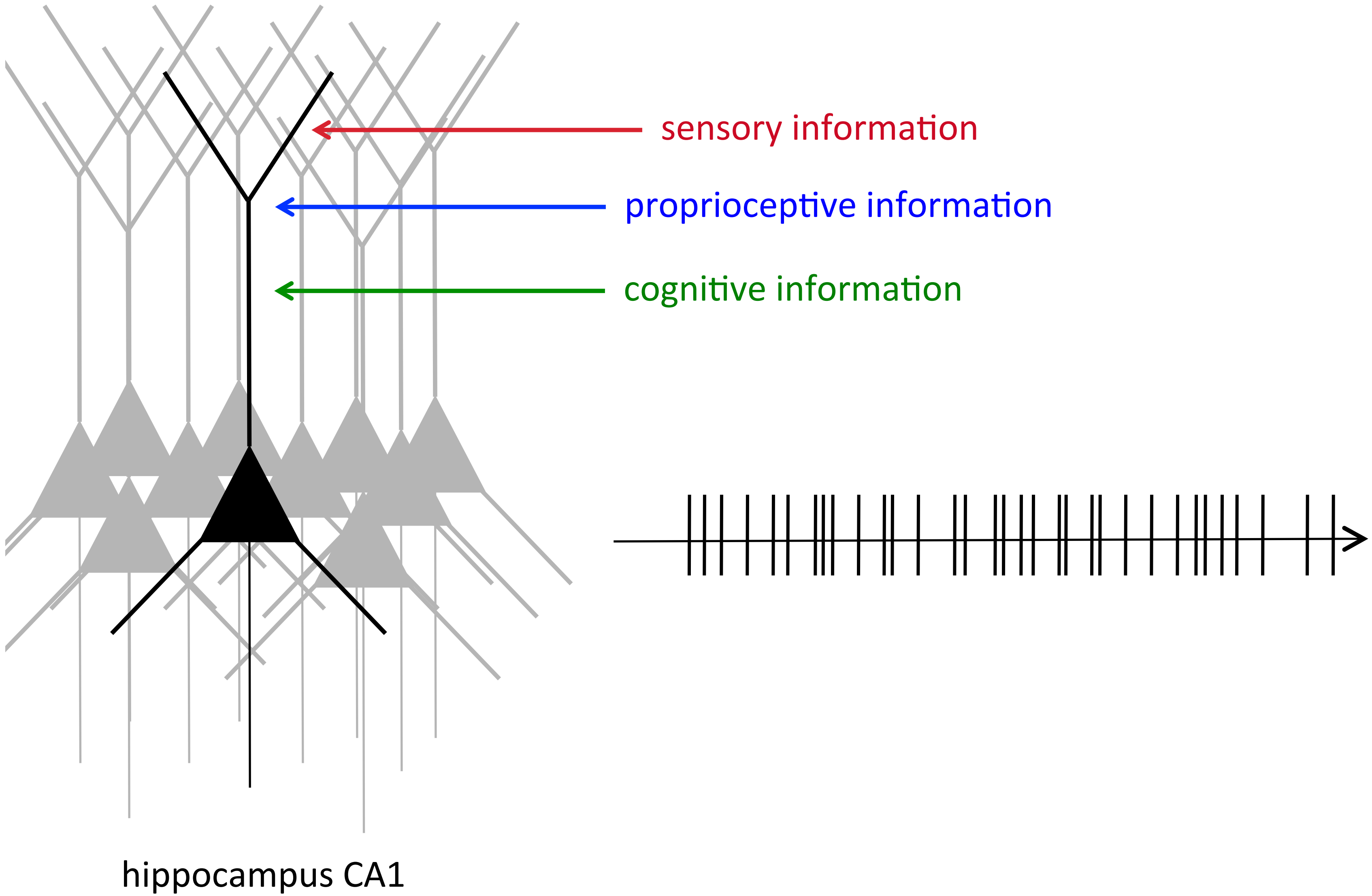


from Amaral DG, Witter MP. (1995) Hippocampal formation. in: Paxinos G, editor. The Rat Nervous System.

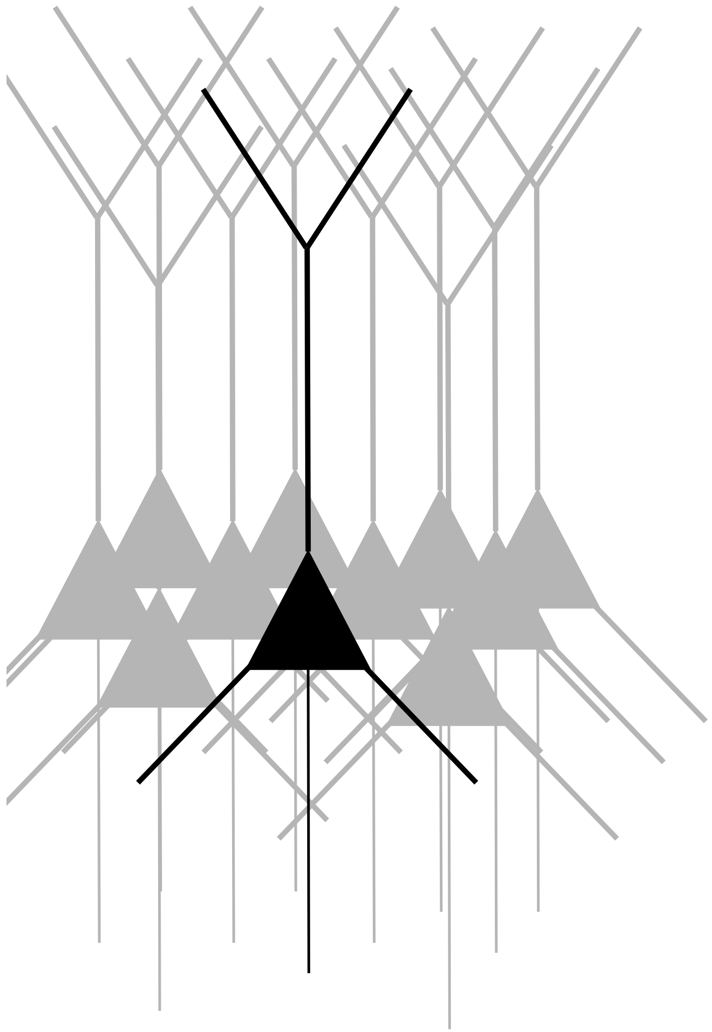
Place cells in the hippocampus

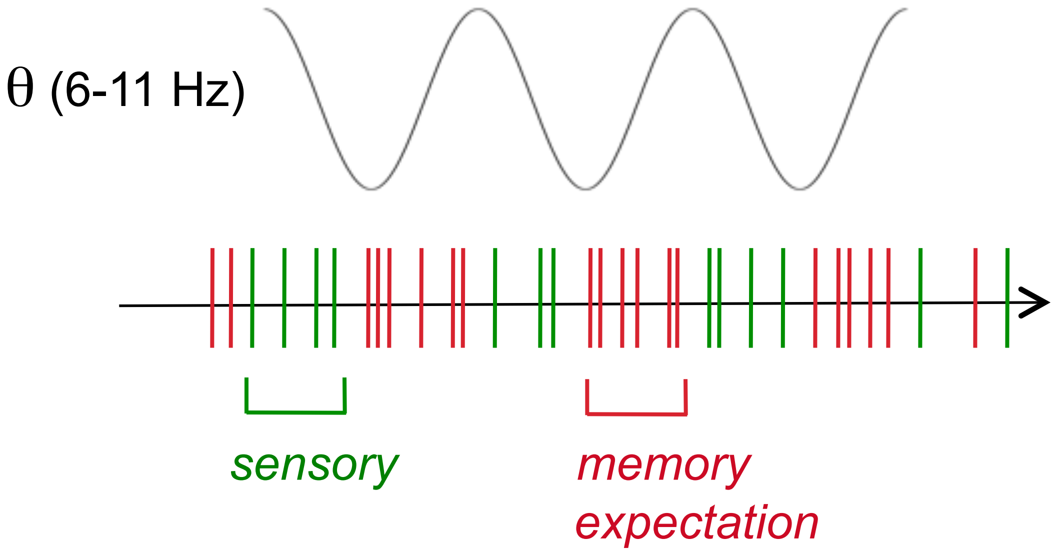



sensory information
proprioceptive information
cognitive information

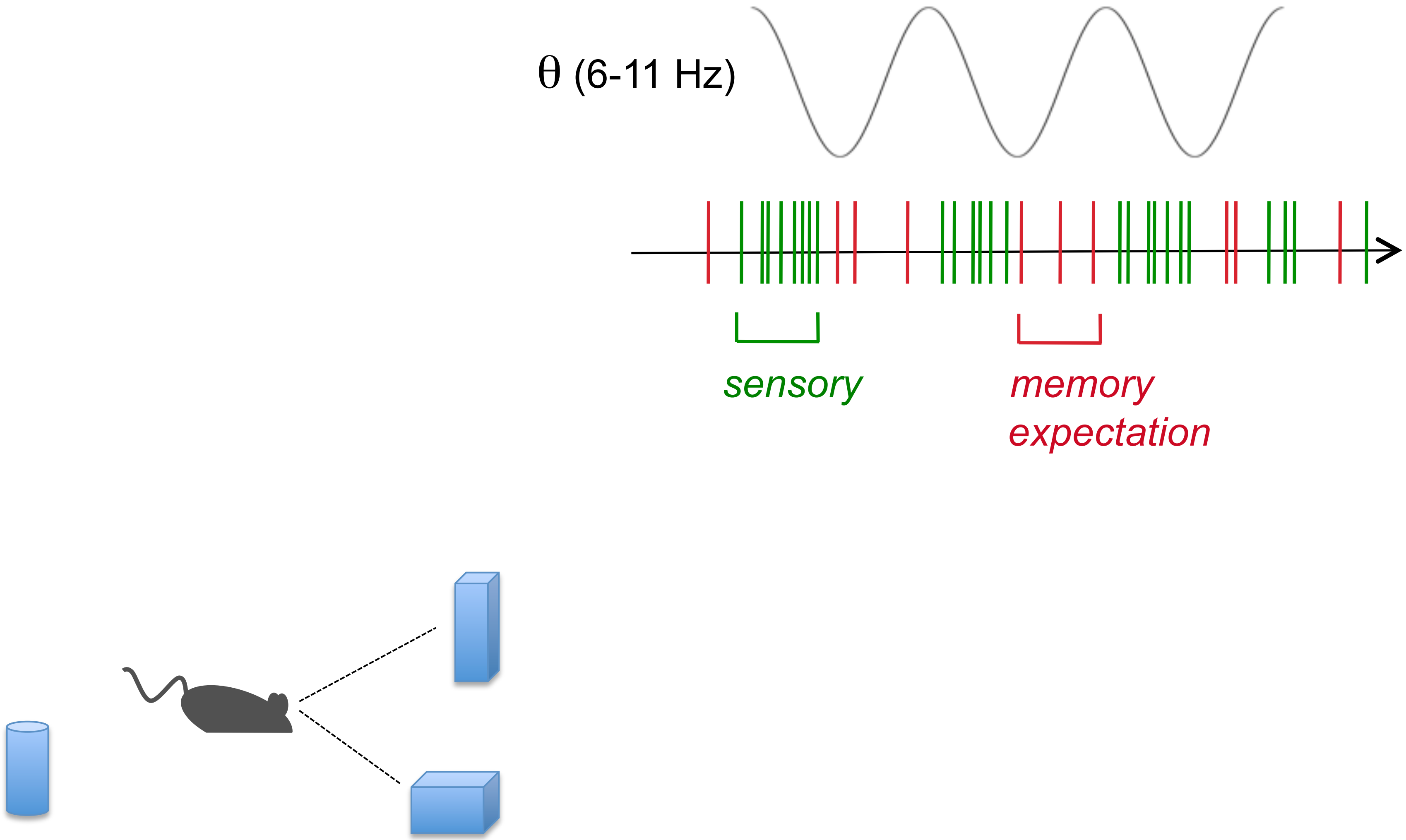


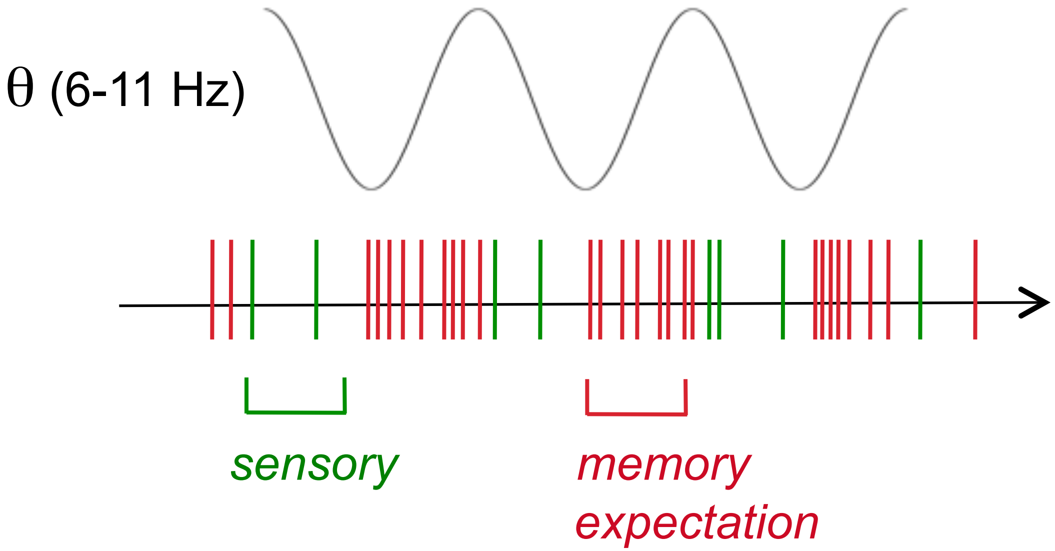
Chance, F.S. (2012) J. Neuroscience 32: 16693-16703.

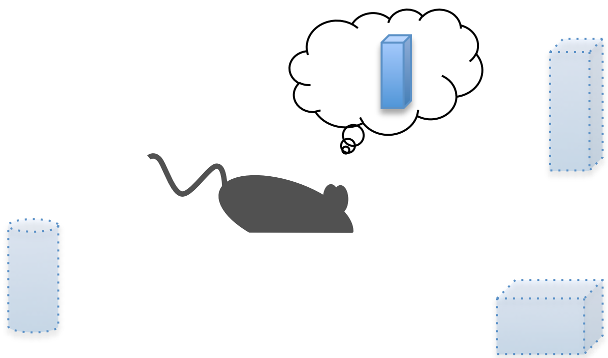




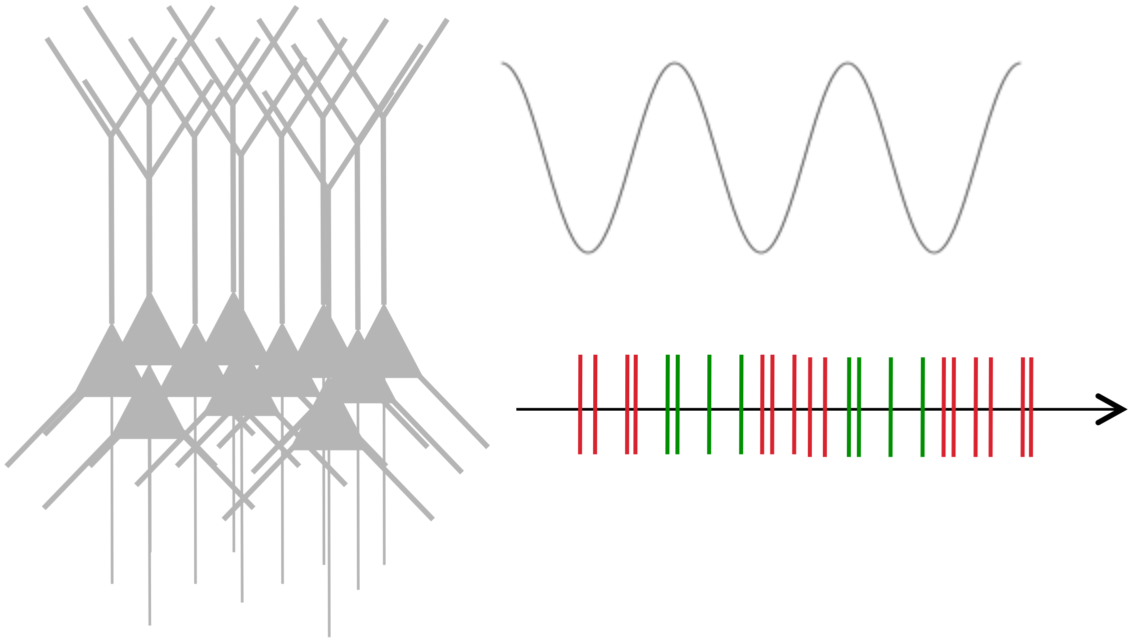
Chance, F.S. (2012) J. Neuroscience 32: 16693-16703.


θ (6-11 Hz)
sensory
memory
expectation

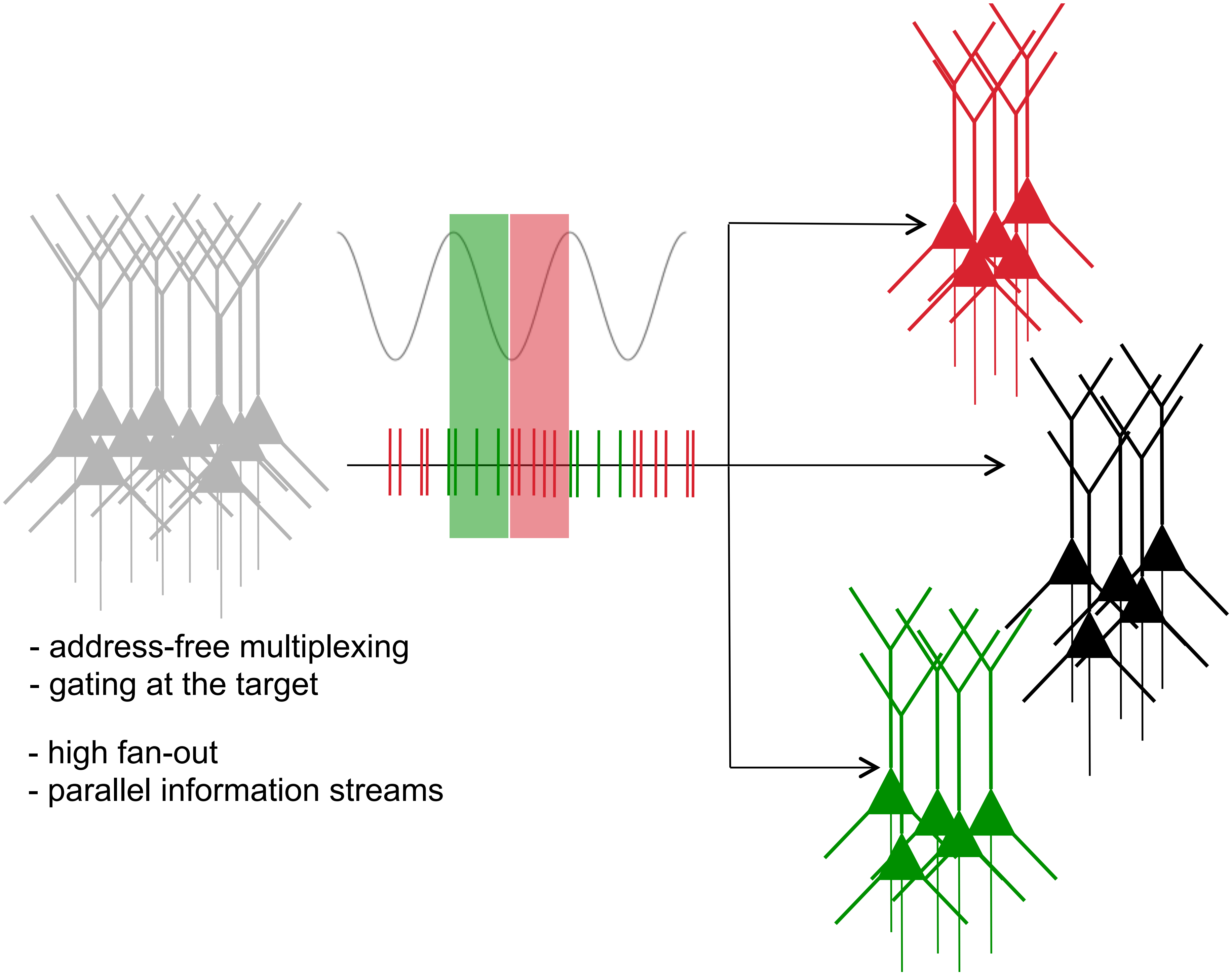




Cabral et al (2014) *Neuron* 81: 402-415.

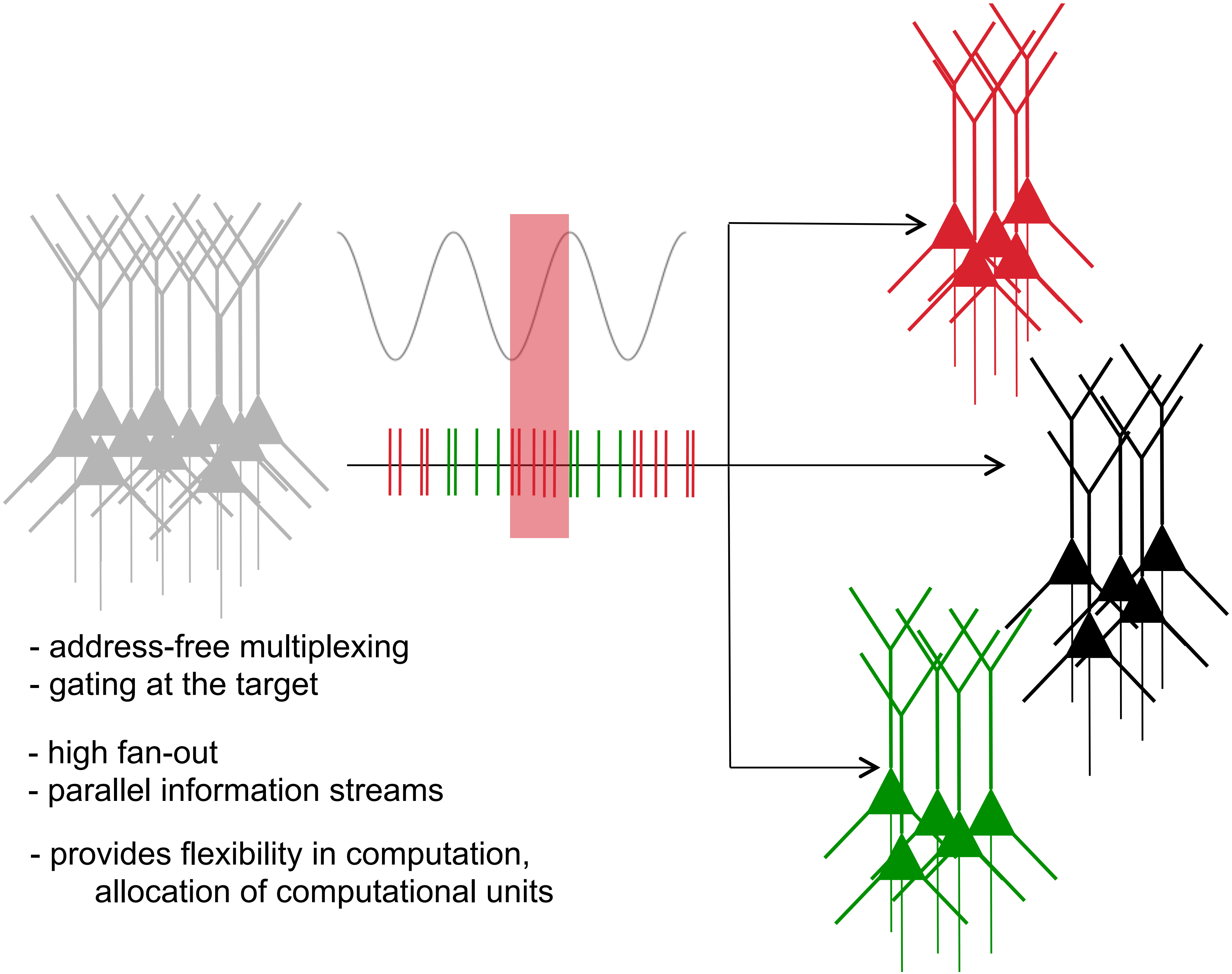


- address-free multiplexing



- address-free multiplexing
- gating at the target

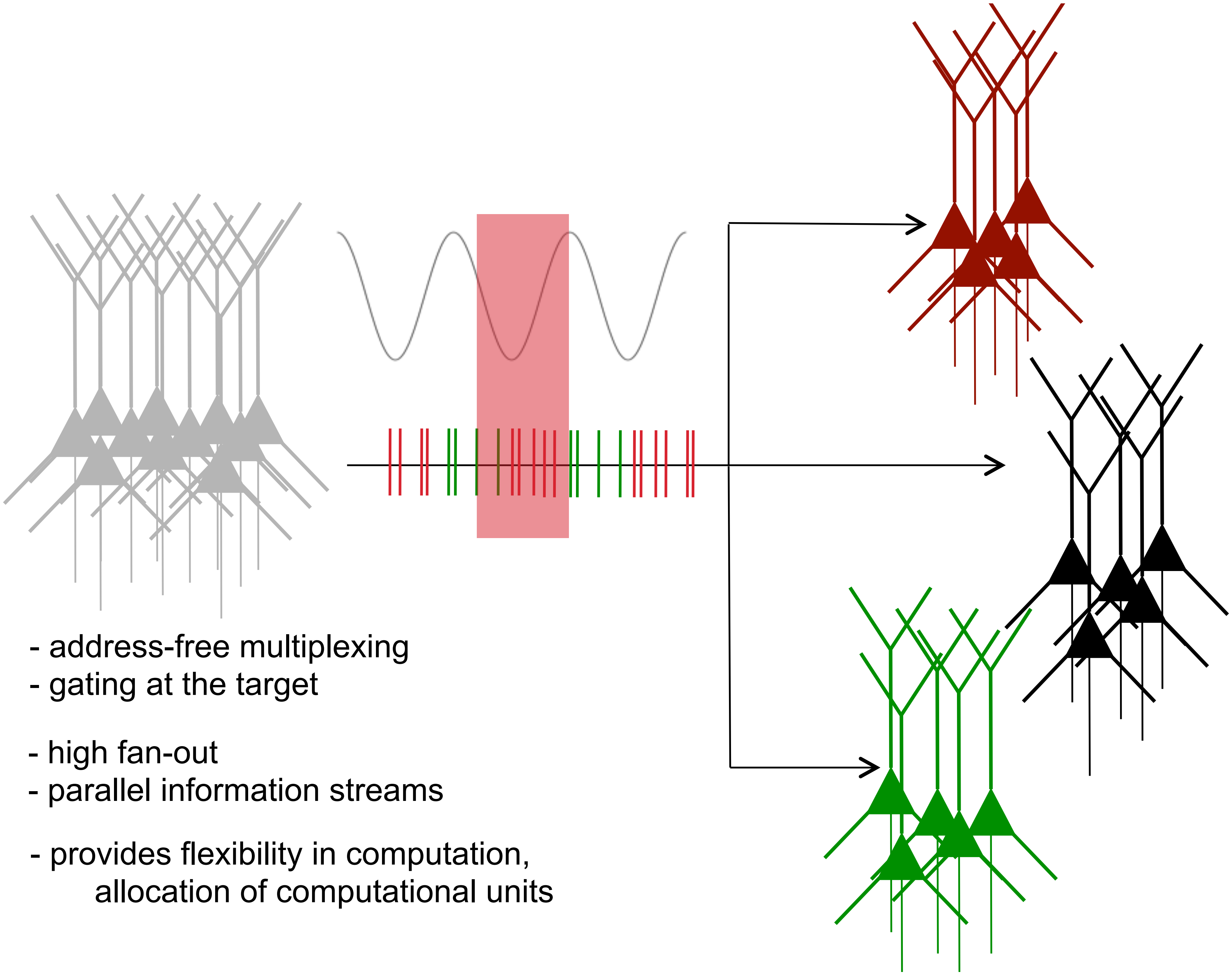
- high fan-out
- parallel information streams



- address-free multiplexing
- gating at the target

- high fan-out
- parallel information streams

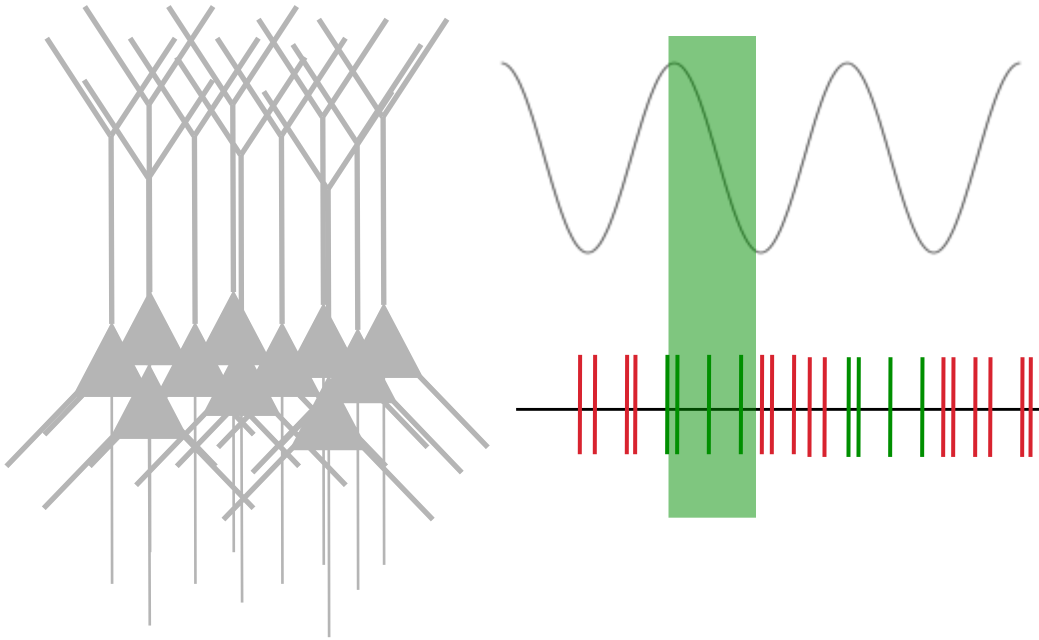
- provides flexibility in computation,
  allocation of computational units

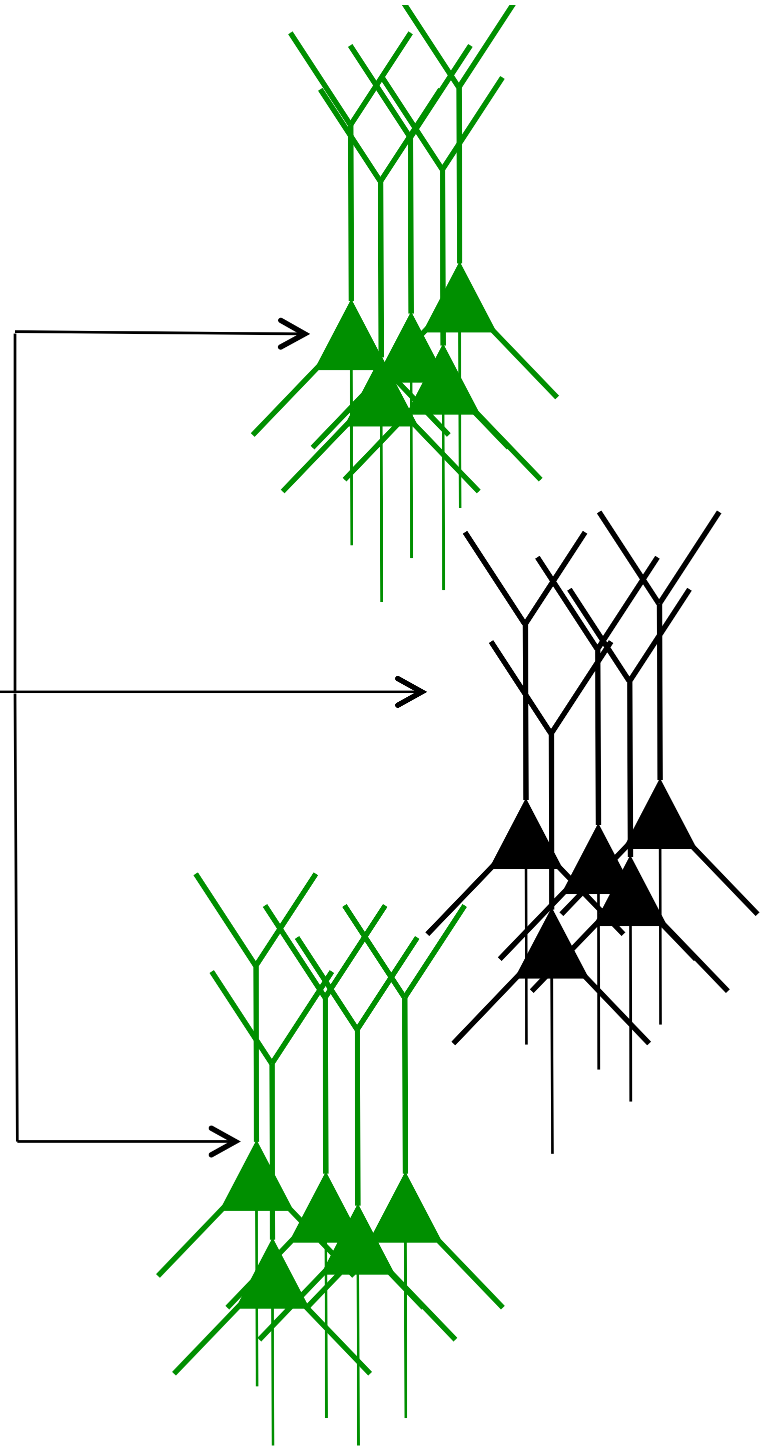


- address-free multiplexing
- gating at the target

- high fan-out
- parallel information streams

- provides flexibility in computation,
  allocation of computational units



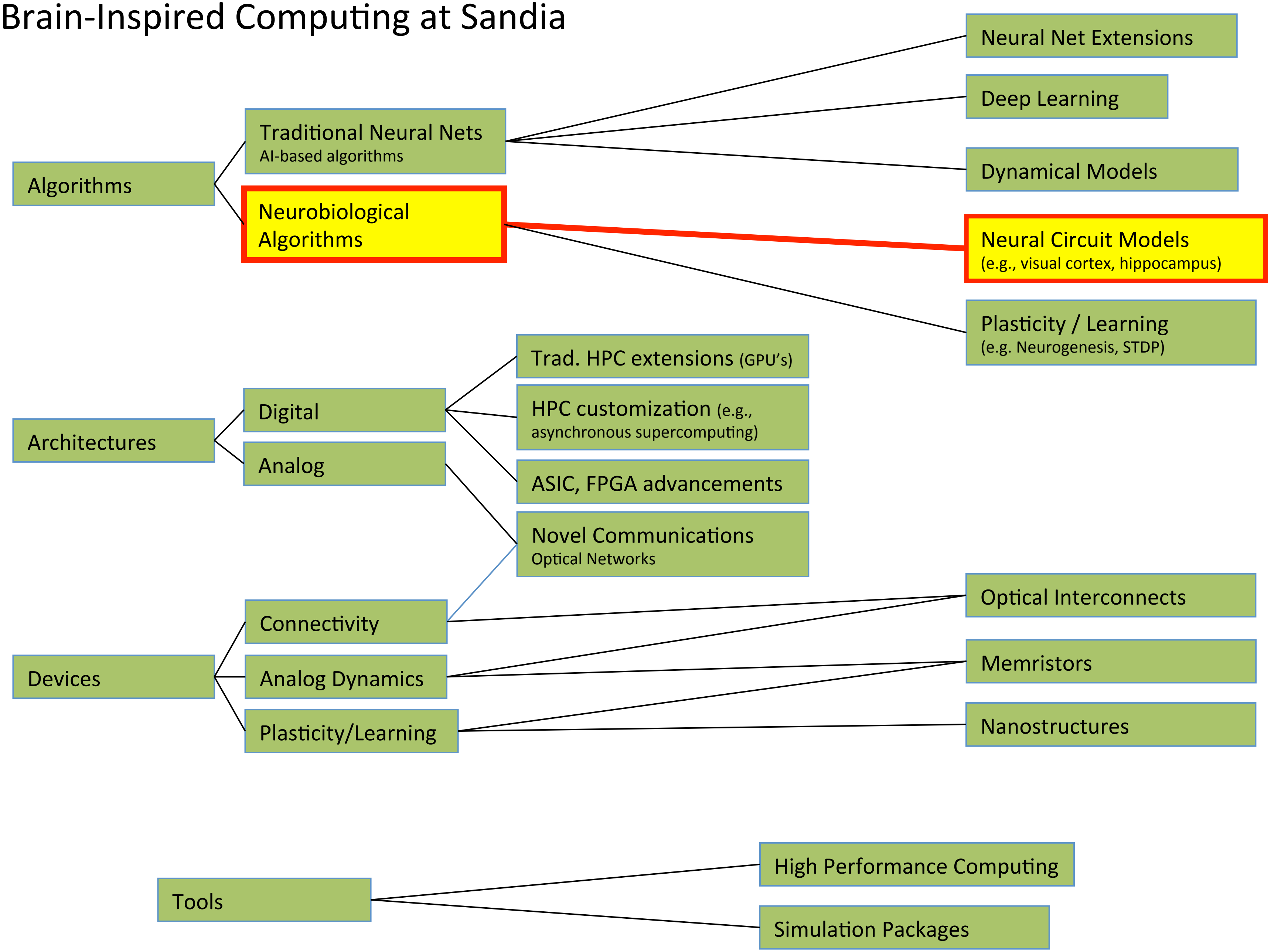


- address-free multiplexing
- gating at the target

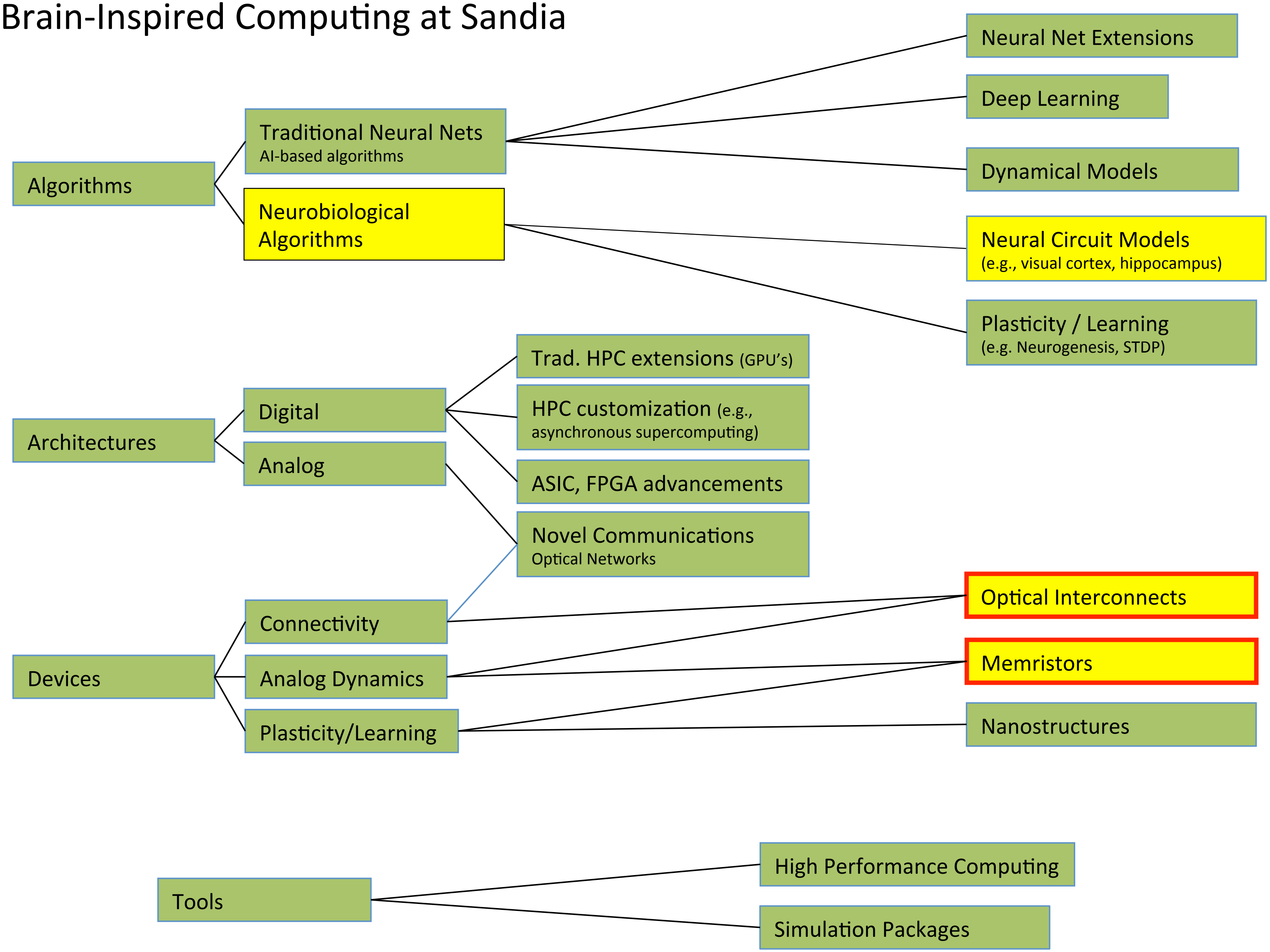
- high fan-out
- parallel information streams

- provides flexibility in computation,
  allocation of computational units

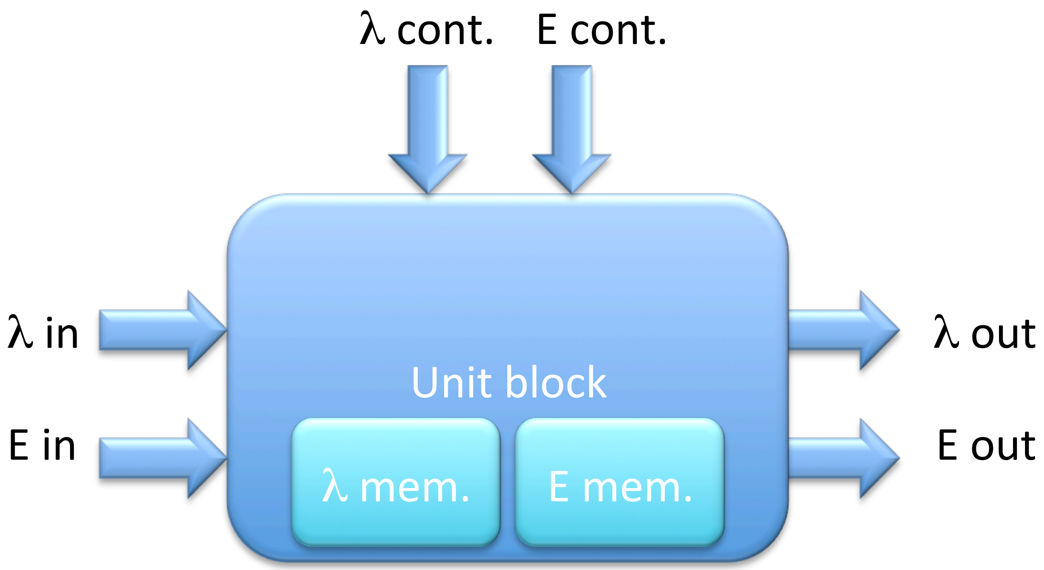
Brain-Inspired Computing at Sandia



Brain-Inspired Computing at Sandia

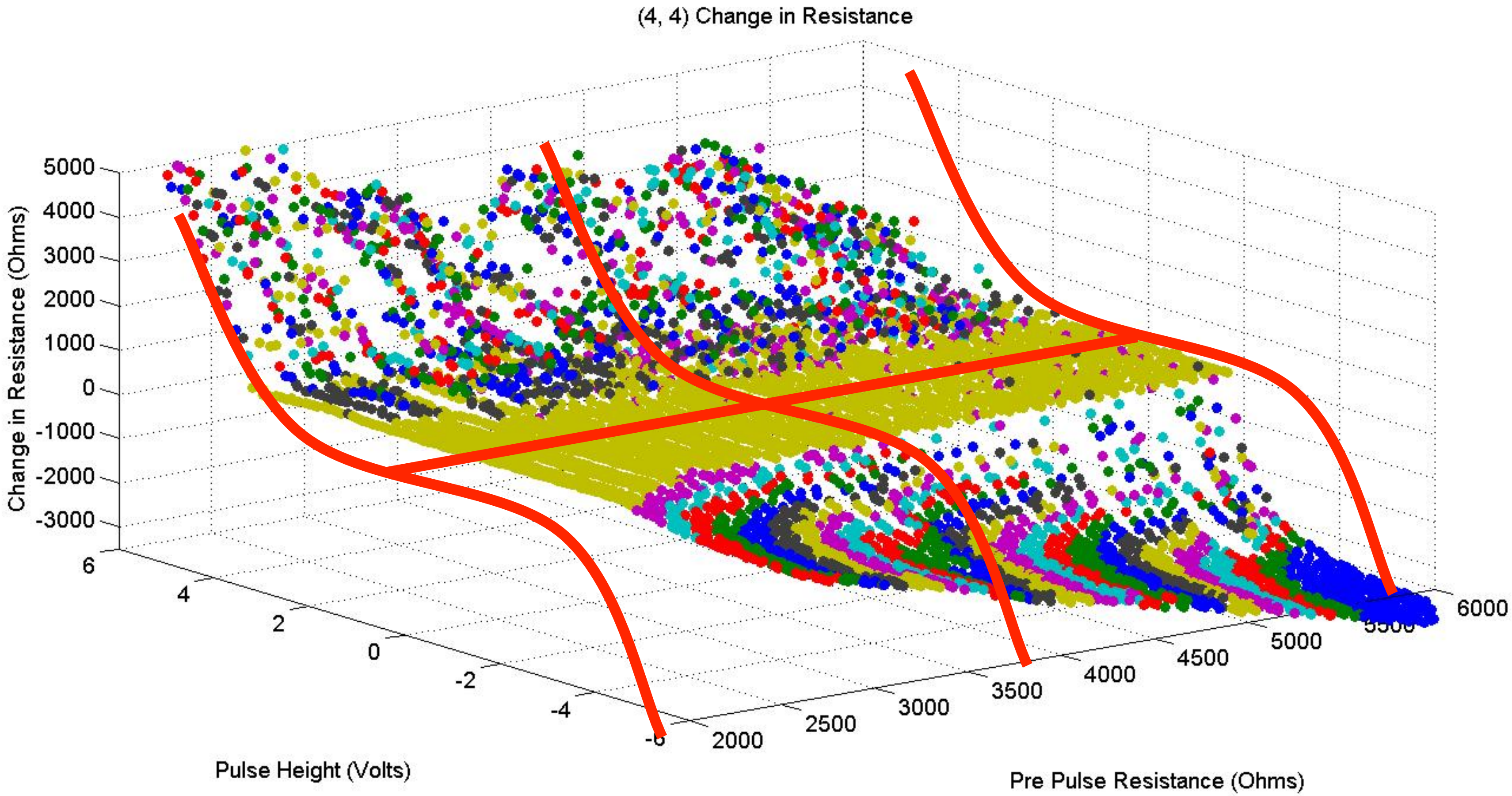


Neural-inspired computational engine: optical interconnect

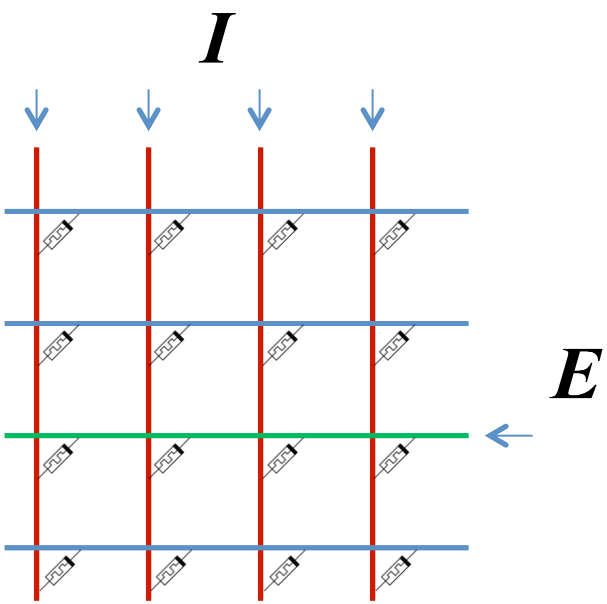


Murat Okandan

Neural-like plasticity: memristor

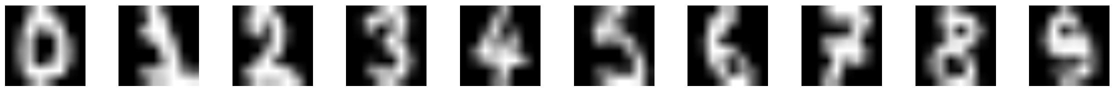


Erik DeBenedictis and Matt Marinella



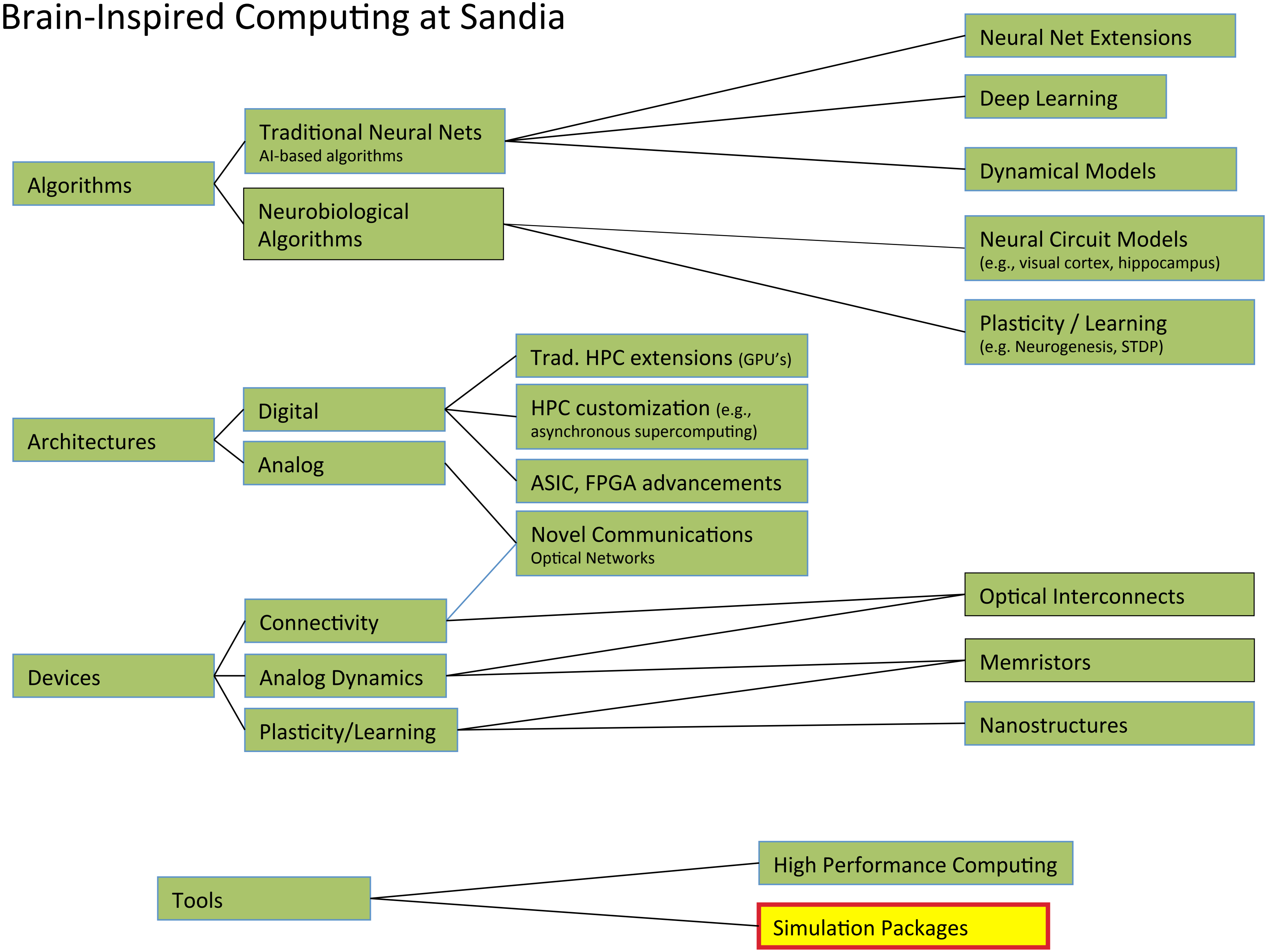
Backprop training

- $dW = \alpha E I^T$ where ***E*** is error vector and ***I*** is input
- Update each row separately so column values can be set appropriately for given error value.
- Test on handwritten digit recognition (MNIST dataset).
- Accuracy = 91.8% (classic methods reach 99.9%)

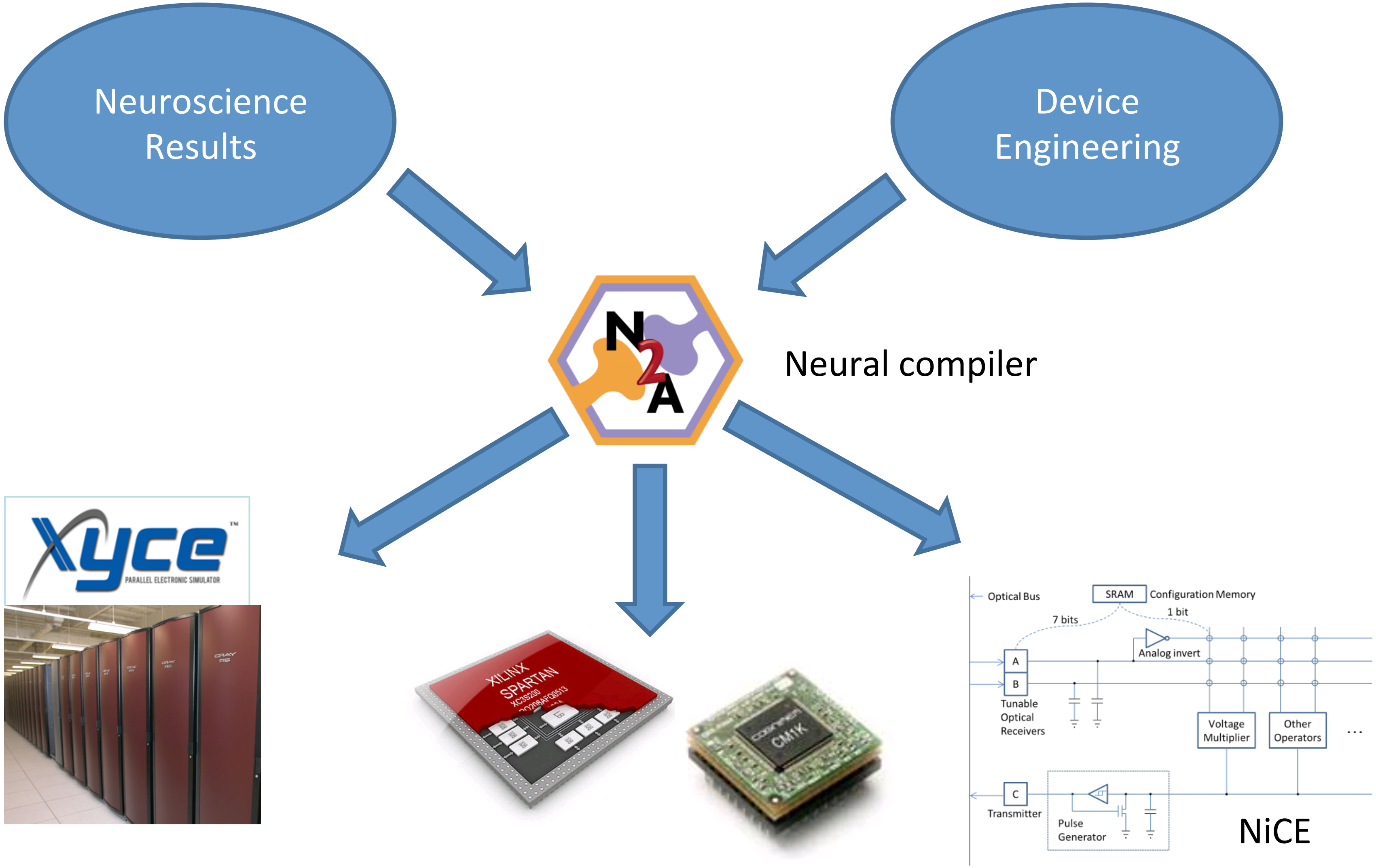


Fred Rothganger, Erik DeBenedictis and Matt Marinella

Brain-Inspired Computing at Sandia

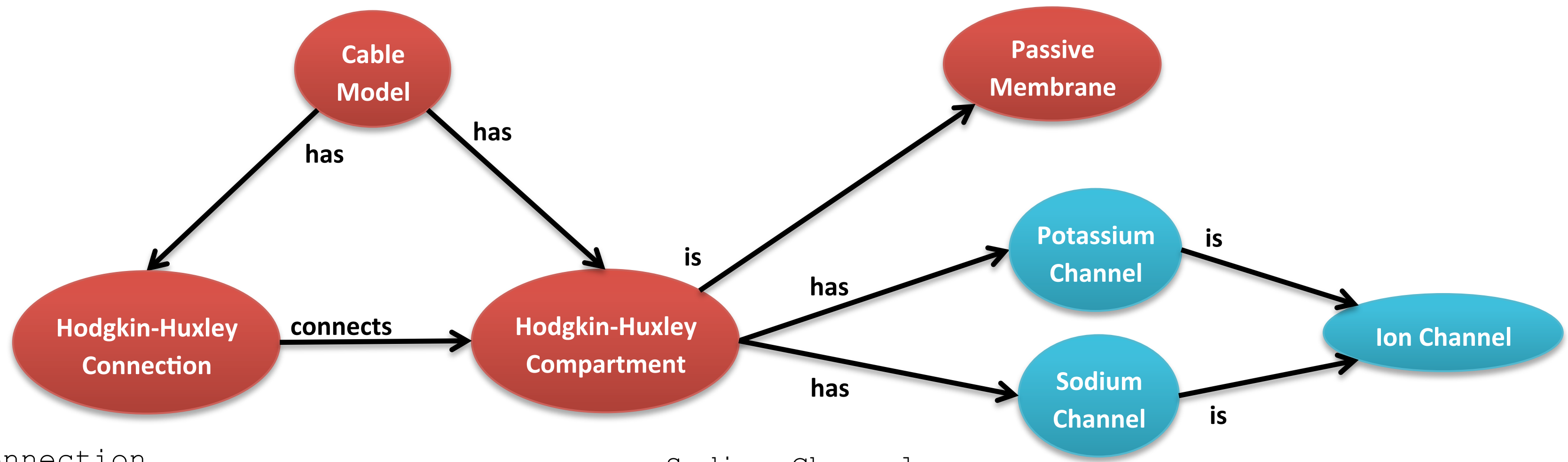


How to program brain-inspired computers?



Fred Rothganger

Example: Hodgkin-Huxley cable model



<u>HH Connection</u>

```
A      = $connect ("HH Compartment")
B      = $connect ("HH Compartment")
A.V' += (B.V - A.V) / R
B.V' += (A.V - B.V) / R
R     = 10
```

<u>HH Compartment</u>

```
parent = $inherit ("Passive Membrane")
K      = $include ("Potassium Channel")
Na     = $include ("Sodium Channel")
```

<u>Sodium Channel</u>

```
parent   = $inherit ("Ion Channel")
I        = G * m^3 * h * (E - V)
m'       = alpha_m * (1 - m) - beta_m * m
h'       = alpha_h * (1 - h) - beta_h * h
alpha_m := (25 - V) / (10 * (exp ((25 - V) / 10) - 1))
beta_m  := 4 * exp (- V / 18)
alpha_h := 0.07 * exp (- V / 20)
beta_h  := 1 / (exp ((30 - V) / 10) + 1)
G        = 120
E        = 115
```

Fred Rothganger

N2A neural-compiler outputs

Xyce

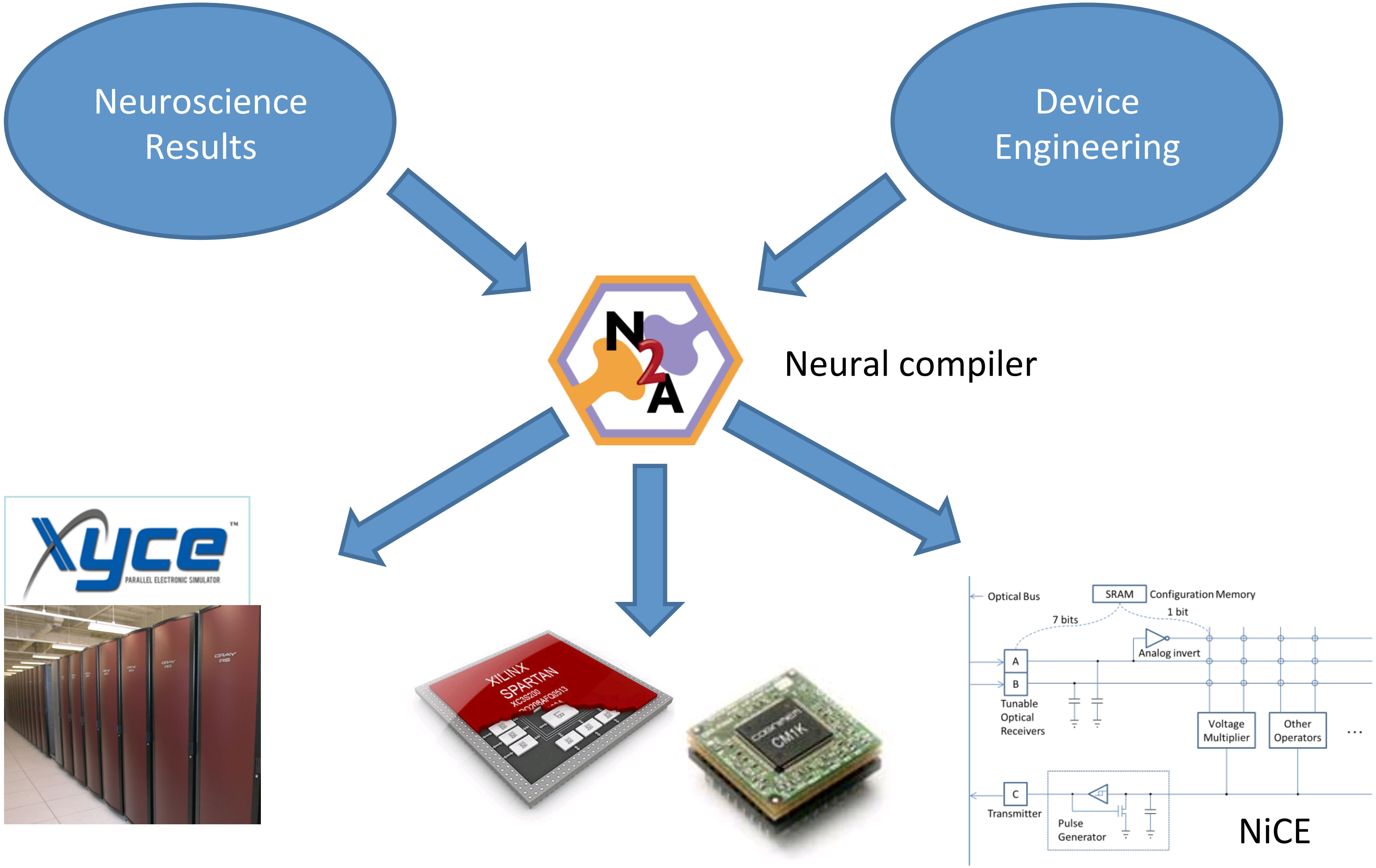
```
* seed:  1380077478872
.tran 0 0.0

* initial condition equations for HHmod

* remaining equations for HHmod
CV_1 V_1_node 0 1
BV_1_equ 0 V_1_node I={ (G_m_1 * (V_rest_1 - V(V_1_node)) + I_inj_1) / C_m_1}
.param I_inj_1 = {10.0}
.param C_m_1 = {1.0}
.param V_rest_1 = {10.613}
.param G_m_1 = {0.3}
CV_2 V_2_node 0 1
BV_2_equ 0 V_2_node I={ (G_m_1 * (V_rest_1 - V(V_2_node)) + I_inj_1) / C_m_1}
CV_3 V_3_node 0 1
BV_3_equ 0 V_3_node I={ (G_m_1 * (V_rest_1 - V(V_3_node)) + I_inj_1) / C_m_1}

* initial condition equations for HHmodHHmod

* remaining equations for HHmodHHmod
BB.V_4_equ 0 V_2_node I={ (V(V_1_node) - V(V_2_node)) / (C_m_1 * R_4)}
.param R_4 = {10.0}
BA.V_4_equ 0 V_1_node I={ (V(V_2_node) - V(V_1_node)) / (C_m_1 * R_4)}
BB.V_5_equ 0 V_3_node I={ (V(V_2_node) - V(V_3_node)) / (C_m_1 * R_4)}
BA.V_5_equ 0 V_2_node I={ (V(V_3_node) - V(V_2_node)) / (C_m_1 * R_4)}

* outputs
.print tran
```

C++

```
class Model : public _Compartment
{
public:
  class HHmod : public _Compartment
  {
  public:
    …
    virtual void update (float _24t, float & _24dt)
    {
      float Na_2ealpha_5fh;
      Na_2ealpha_5fh = 0.07f * exp(-V / 20.0f);
      float Na_2ealpha_5fm;
      Na_2ealpha_5fm = (25.0f - V) / (10.0f * (exp((25.0f - V) / 10.0f) - 1.0f));
      float K_2ealpha_5fn;
      K_2ealpha_5fn = (10.0f - V) / (100.0f * (exp((10.0f - V) / 10.0f) - 1.0f));
      float Na_2ebeta_5fh;
      Na_2ebeta_5fh = 1.0f / (exp((30.0f - V) / 10.0f) + 1.0f);
      float Na_2ebeta_5fm;
      Na_2ebeta_5fm = 4.0f * exp(-V / 18.0f);
      float K_2ebeta_5fn;
      K_2ebeta_5fn = 0.125f * exp(-V / 80.0f);
      _nextV_27 += (0.3f * (10.613f - V) + I_5finj) / 1.0f + K_2eI / 1.0f + Na_2eI / 1.0f;
      Na_2eh_27 = Na_2ealpha_5fh * (1.0f - Na_2eh) - Na_2ebeta_5fh * Na_2eh;
      Na_2eI = 120.0f * pow (Na_2em, 3.0f) * Na_2eh * (115.0f - V);
      K_2eI = 36.0f * pow (K_2en, 4.0f) * (-12.0f - V);
      Na_2em_27 = Na_2ealpha_5fm * (1.0f - Na_2em) - Na_2ebeta_5fm * Na_2em;
      K_2en_27 = K_2ealpha_5fn * (1.0f - K_2en) - K_2ebeta_5fn * K_2en;
      if (_24index == 0.0f)
        I_5finj = 10.0f;
    }
    …
  };
};
```

Fred Rothganger

How to program brain-inspired computers?



Fred Rothganger

Brain-Inspired Computing at Sandia