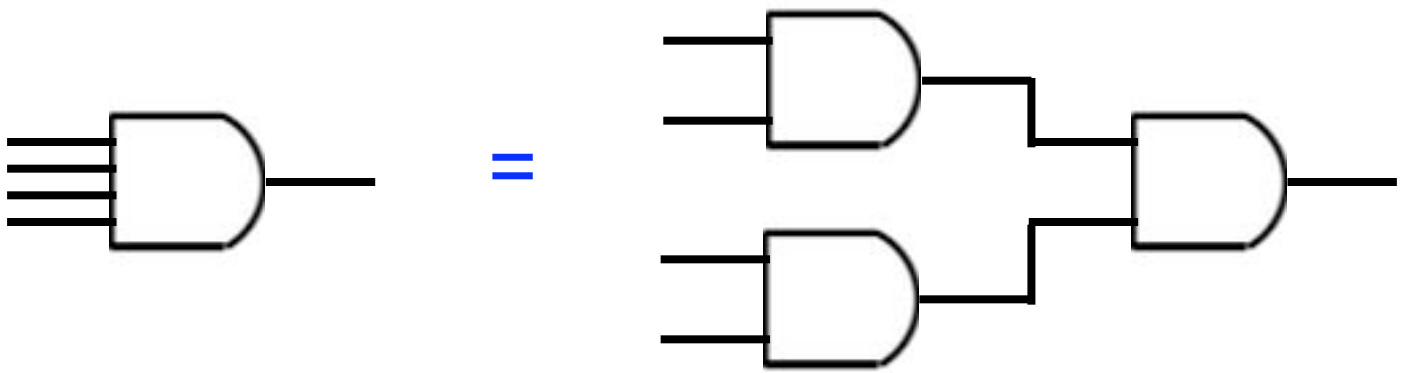
Recursion: N-input AND gate

A 2N-input AND gate is built from a pair of N-input AND gates and a single 2-input AND



CSL

Expressing this in CAST

```
define AndN(int N)(node[N] in; node out)
{
  [N=1 -> out = in[0];]
  [N=2 -> And2()(in[0], in[1], out);]
  [N>2 ->
    int N2 = N/2;
    And2 a2(,,out);
    AndN(N2)(in[0..N2-1], a2.a);
    AndN(N-N2)(in[N2..N-1], a2.b);
  ]
}
```

CAST conditions

(cases for both N=1 and N=2 allow use of AndN(X) even if X is not a power of 2)

What if N is not a power of 2?

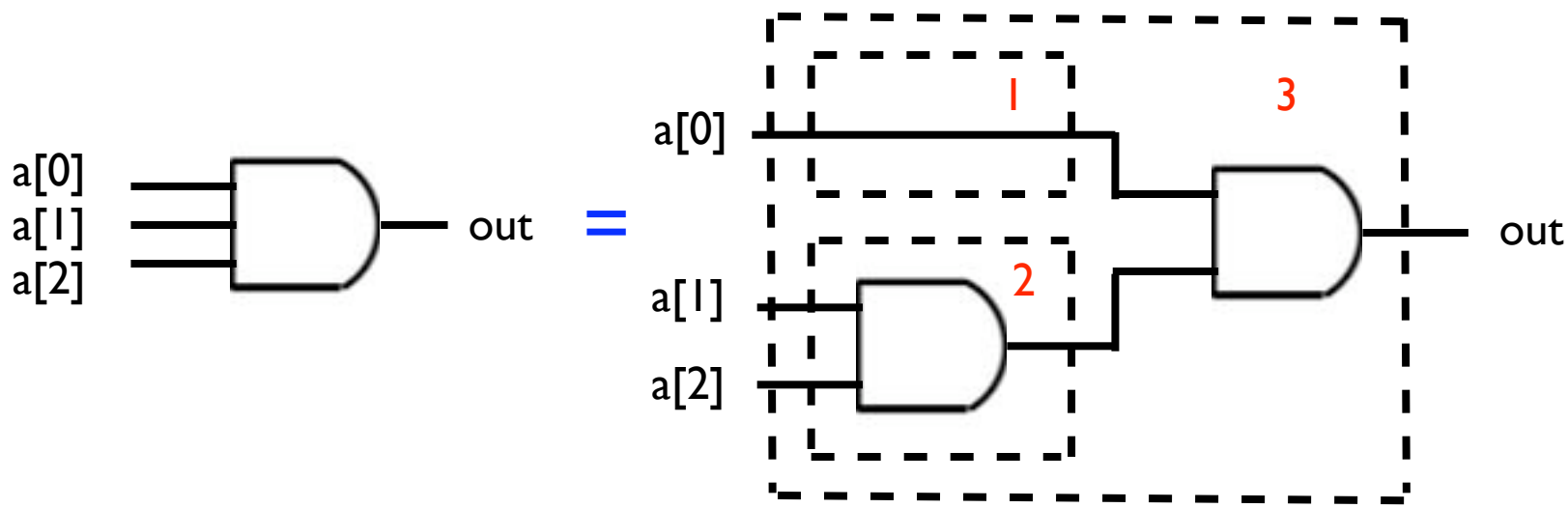
Example:

AndN(3)(...)

Note N2 = 3/2 = 1 and N-N2 = 3-1 = 2

So this instantiates

AndN(1)(...) and AndN(2)(...)



CSL

A More Sophisticated Example

An L-bit decoder with N outputs ...

Inputs:

node[L] in	L-bit input
node enb	output is 0 unless enb is TRUE

Outputs:

node[N] out	N-bit output:
	out[i] is set iff i = value of in
	=> L = log(N)

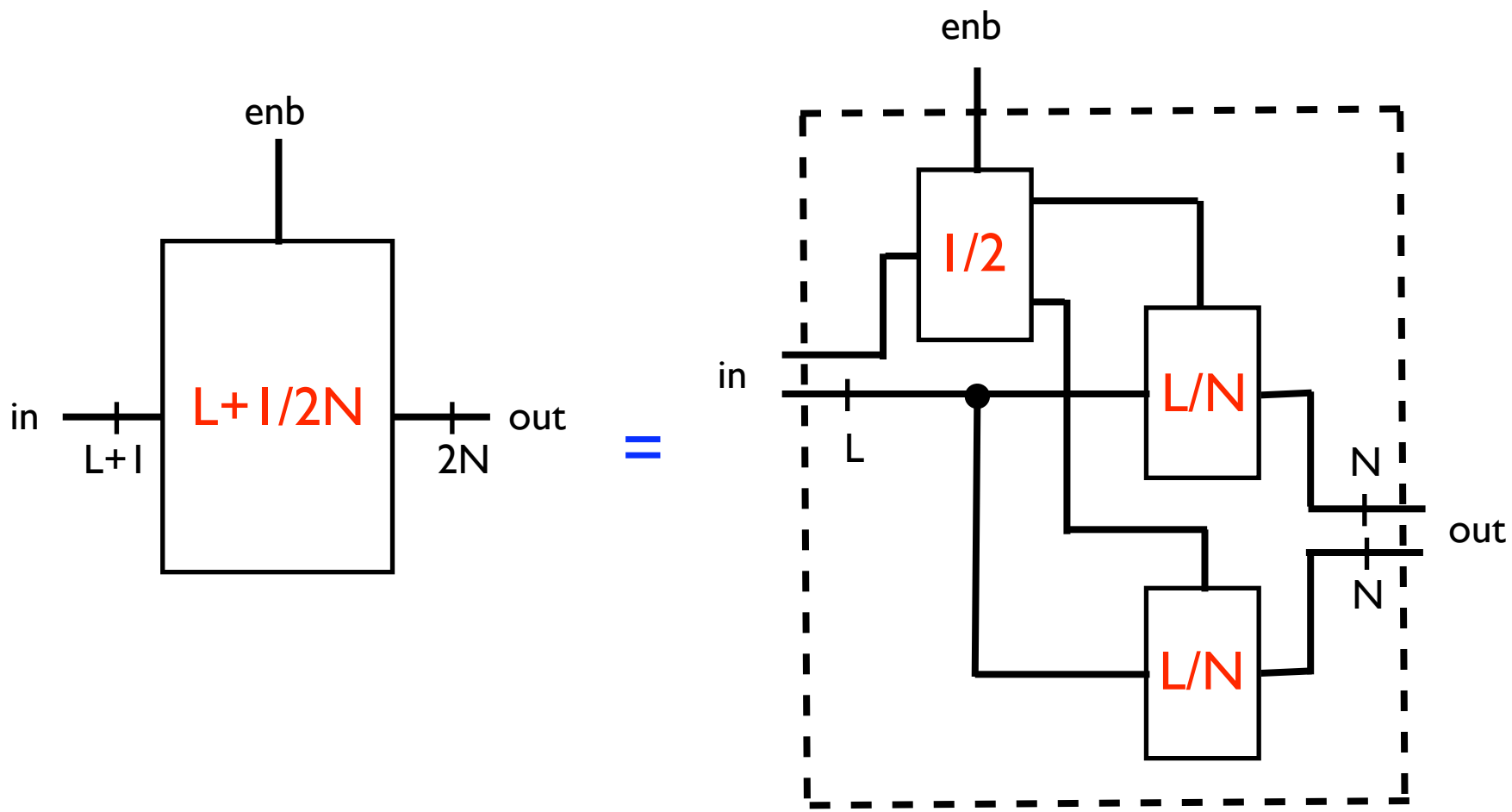
Note this makes no sense for (L=0, N=1)!

define DLN(int L; int N)(node[L] in; node enb; node[N] out)
{ ... }

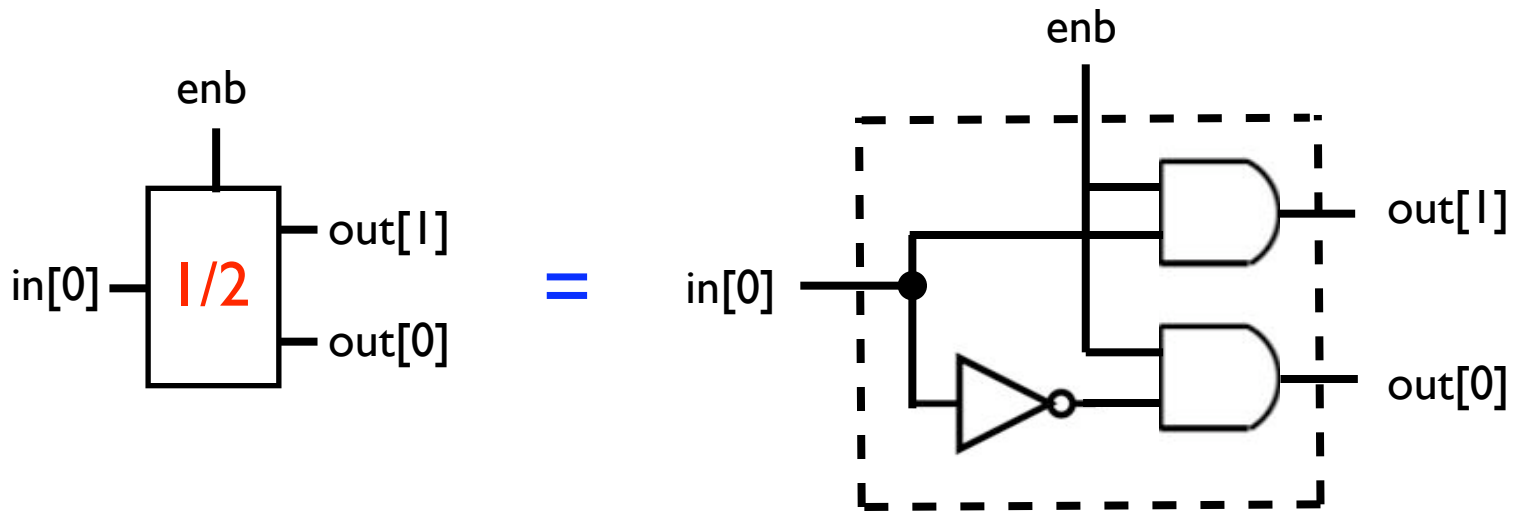
CSL

Recursive Decoder Definition

Text



Decoder Definition - Basis



CSL

Expressing this in CAST

This assumes N is exactly 2**L ...

```
define DLN(int L; int N)(node[L] in; node enb; node[N] out)
{
  [L=1 ->
    node _in0;  Inv()(in[0], _in0);
    And2()(enb, _in0, out[0]);
    And2()(enb, in0, out[1]);
  ]
  [L>1 ->
    DLN(1,2) sel(in[L-1], enb,);
    DLN(L-1,N/2)(in[0..L-2], sel.out[0], out[0..N/2-1]);
    DLN(L-1,N/2)(in[0..L-2], sel.out[1], out[N/2..N-1]);
  ]
}
```

CSL

The Bottom Line ...

Use recursion in CAST to describe large circuits
if they can be defined by "divide-and-conquer"
or as tree-like structures.

Examples:

N-input And, OR, ...
N-input Mux, N-output decoder
N-bit Carry-lookahead adder, incrementer, ...
Fast N-bit zero or equality test
N-bit shift register with variable shift amount

... and a host of others!