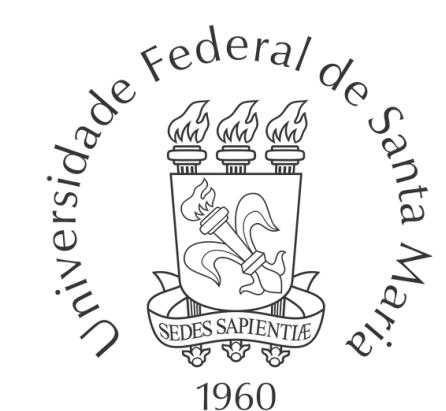


Trabalho de Construção de Circuito para Cálculo de Raiz Quadrada - Versão 2

Vinícius Schultz



Universidade Federal de Santa Maria (UFSM)
2025

MUDANÇAS NO CÓDIGO

CÓDIGO DA VERSÃO 1

```
int main(){
    int input = 0;
    int root = 0;
    int square = 1;
    int ready = 1;

    input = 49;

    while(!(input-square < 0)){
        root = root+1;
        square = square + (root<<1) + 1;
    }
    ready = 0;
}
```

MUDANÇAS NO CÓDIGO

CÓDIGO DA VERSÃO 1

```
int main(){
    int input  = 0;
    int root   = 0;
    int square = 1;
    int ready  = 1;

    input = 49;

    while(!(input-square < 0)){
        root  = root+1;
        square = square + (root<<1) + 1;
    }
    ready = 0;
}
```

DEPENDÊNCIA DE DADOS

MUDANÇAS NO CÓDIGO

```
int main(){
    int input = 0;
    int root = 1;
    int square = 0;
    int ready = 1;

    input = 49;

    while(!(input-square < 0)){
        square = square + (root<<1) - 1;
        root = root+1;
    }
    root = root + 1;
    ready = 0;
}
```

MUDANÇAS NO CÓDIGO

```
int main(){
    int input = 0;
    int root = 0;
    int square = 0;
    int ready = 1;

    input = 49;
    root = root + 1;

    while(!(input-square < 0)){
        square = square + (root<<1) - 1;
        root = root+1;
    }
    root = root + 1;
    ready = 0;
}
```

MUDANÇAS NO CÓDIGO

CÓDIGO DA VERSÃO 1

```
int main(){
    int input  = 0;
    int root   = 0;
    int square = 1;
    int ready  = 1;

    input = 49;

    while(!(input-square < 0)){
        root  = root+1;
        square = square + (root<<1) + 1;
    }
    ready = 0;
}
```

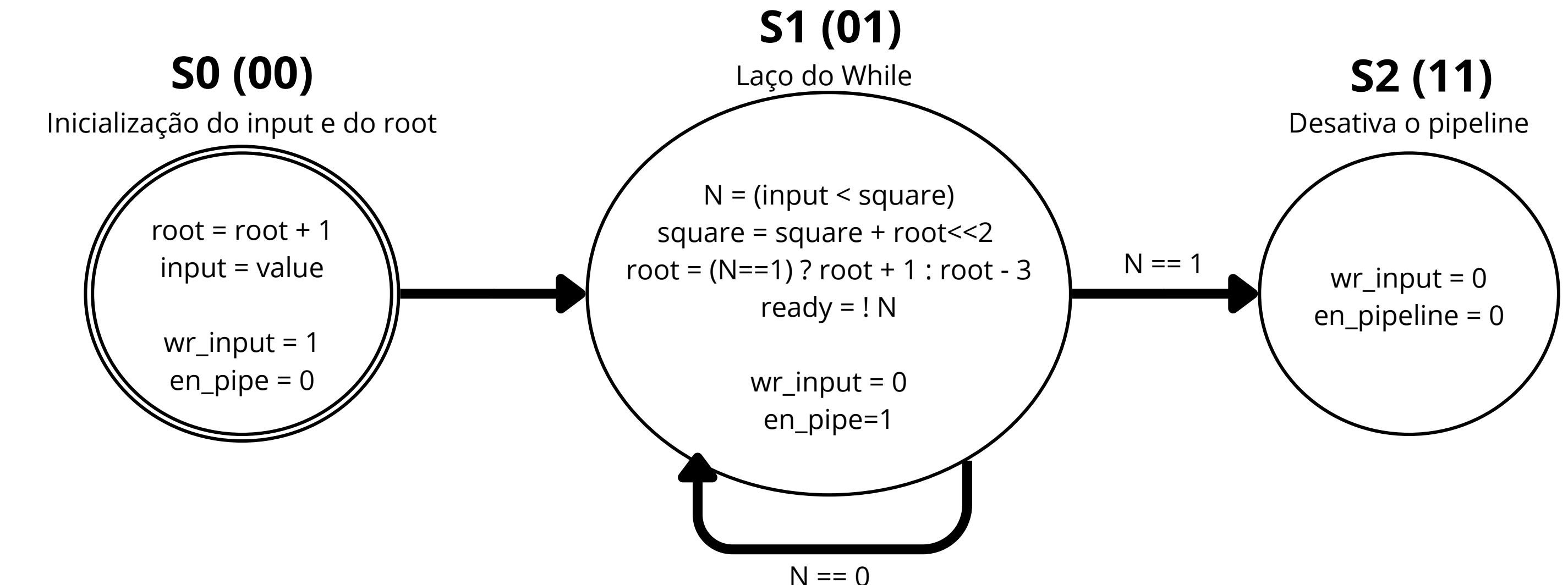
CÓDIGO DA VERSÃO 2

```
int main(){
    int input  = 0;
    int root   = 0;
    int square = 0;
    int ready  = 1;

    input = 49;
    root  = root + 1;

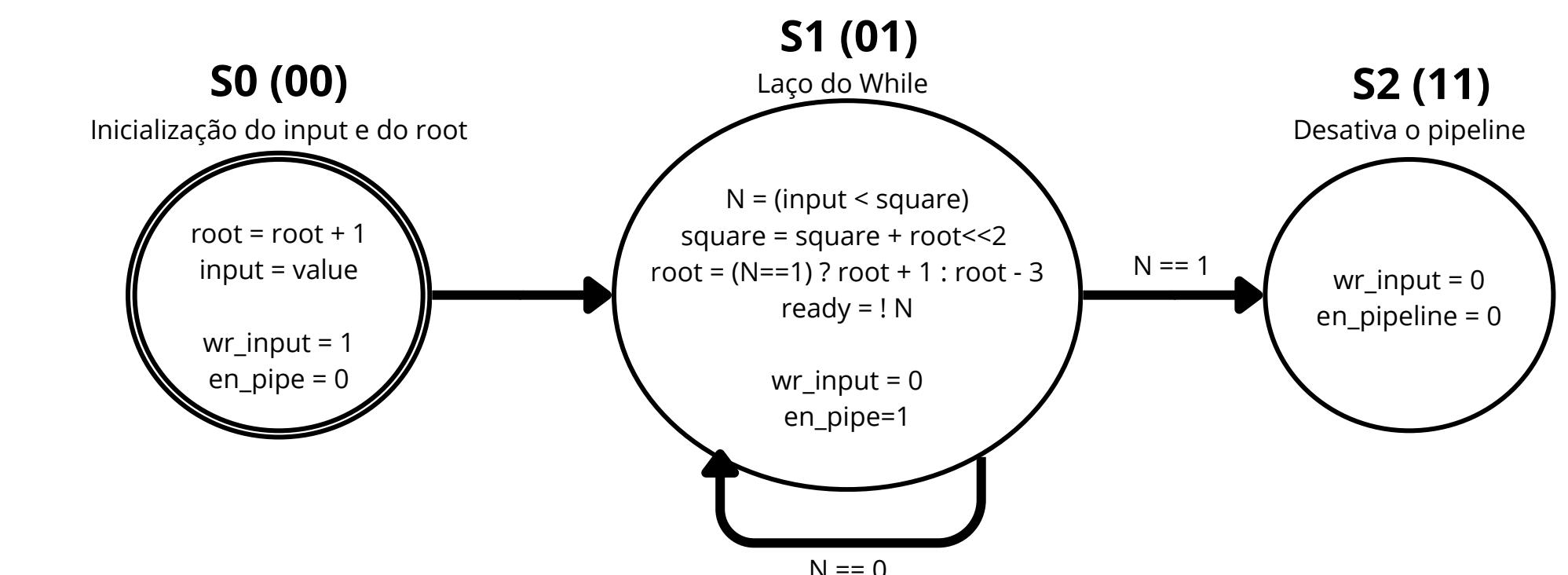
    while(!(input-square < 0)){
        square = square + (root<<1) - 1;
        root  = root+1;
    }
    root = root + 1;
    ready = 0;
}
```

MÁQUINA DE ESTADOS

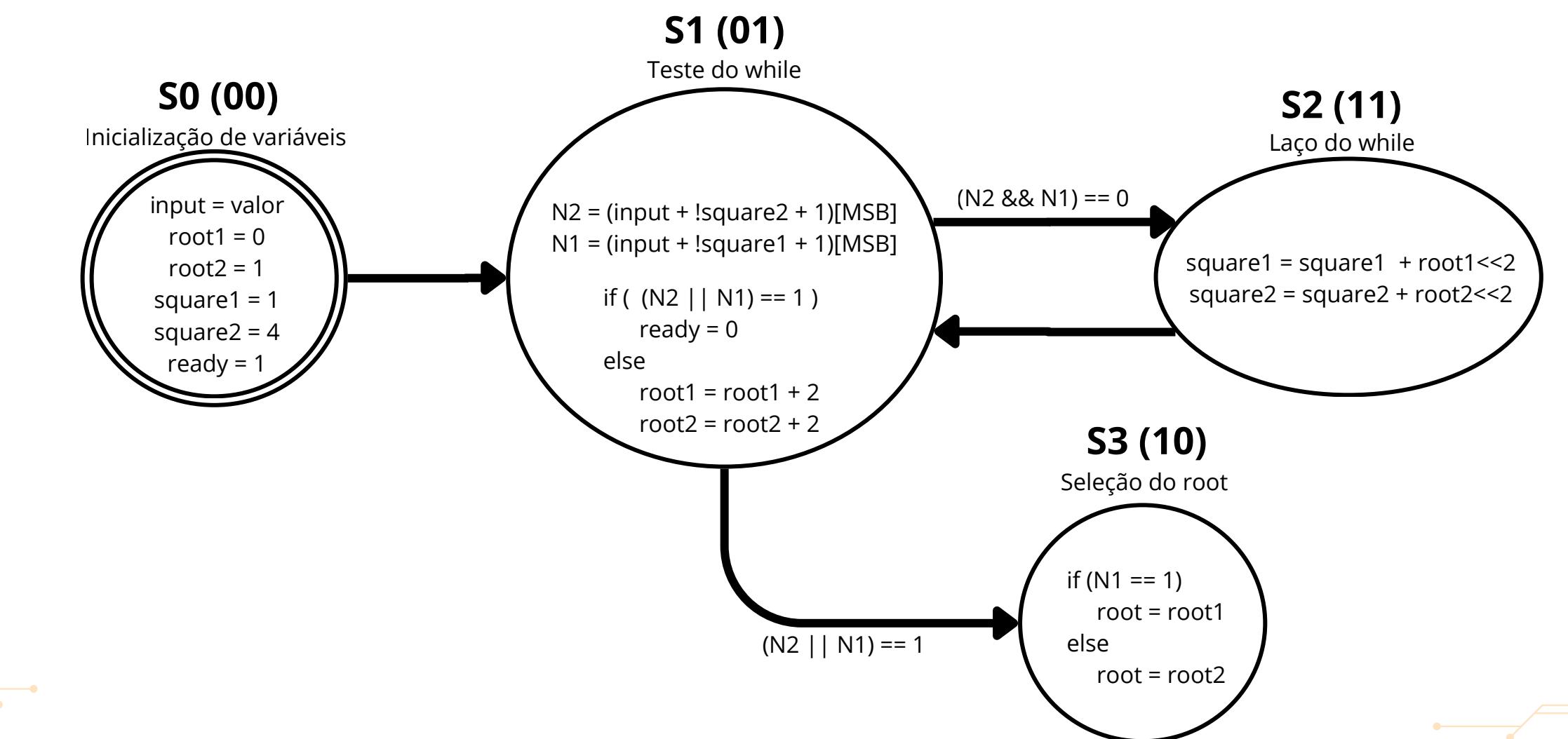


MÁQUINA DE ESTADOS

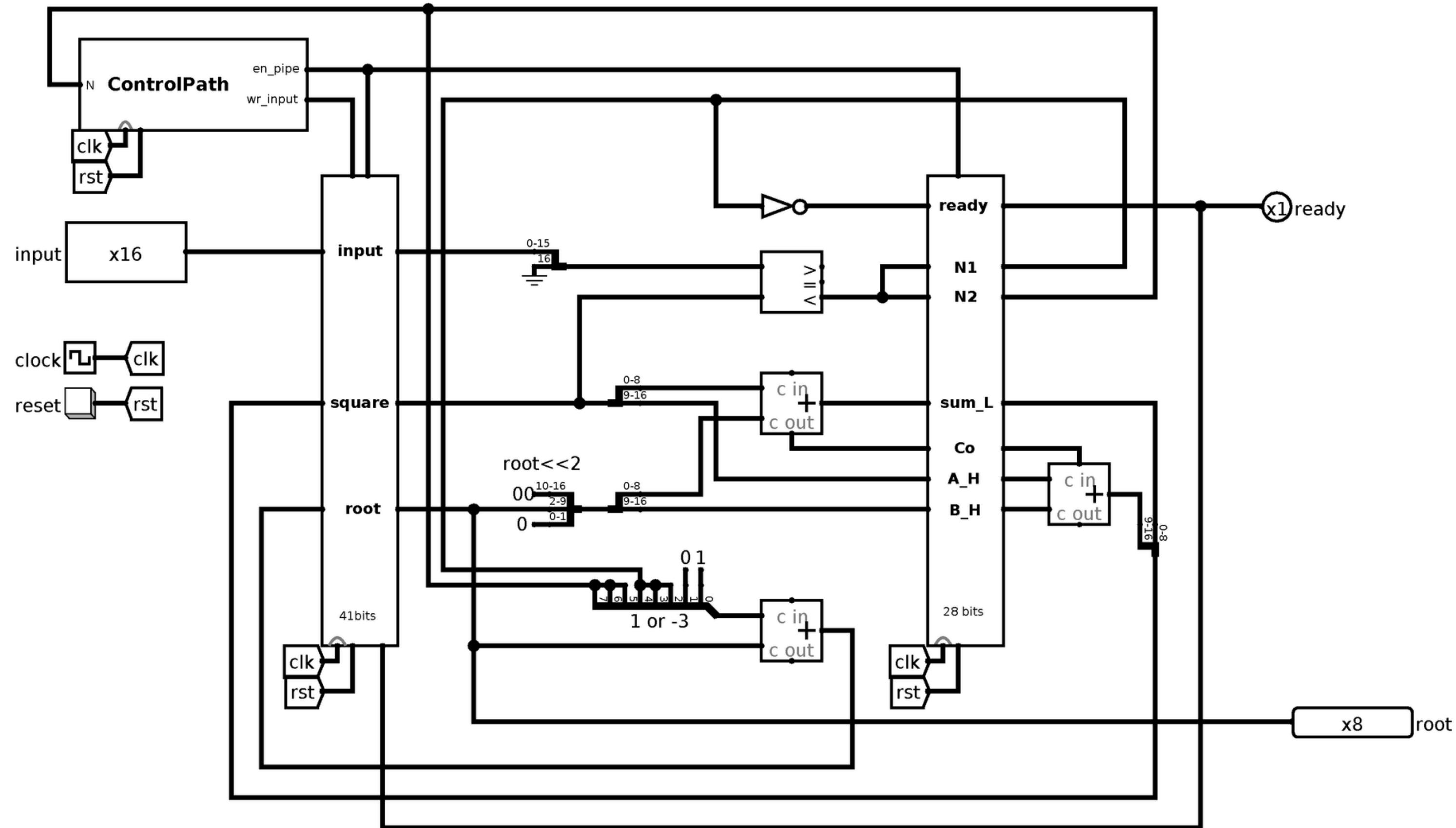
Versão 2



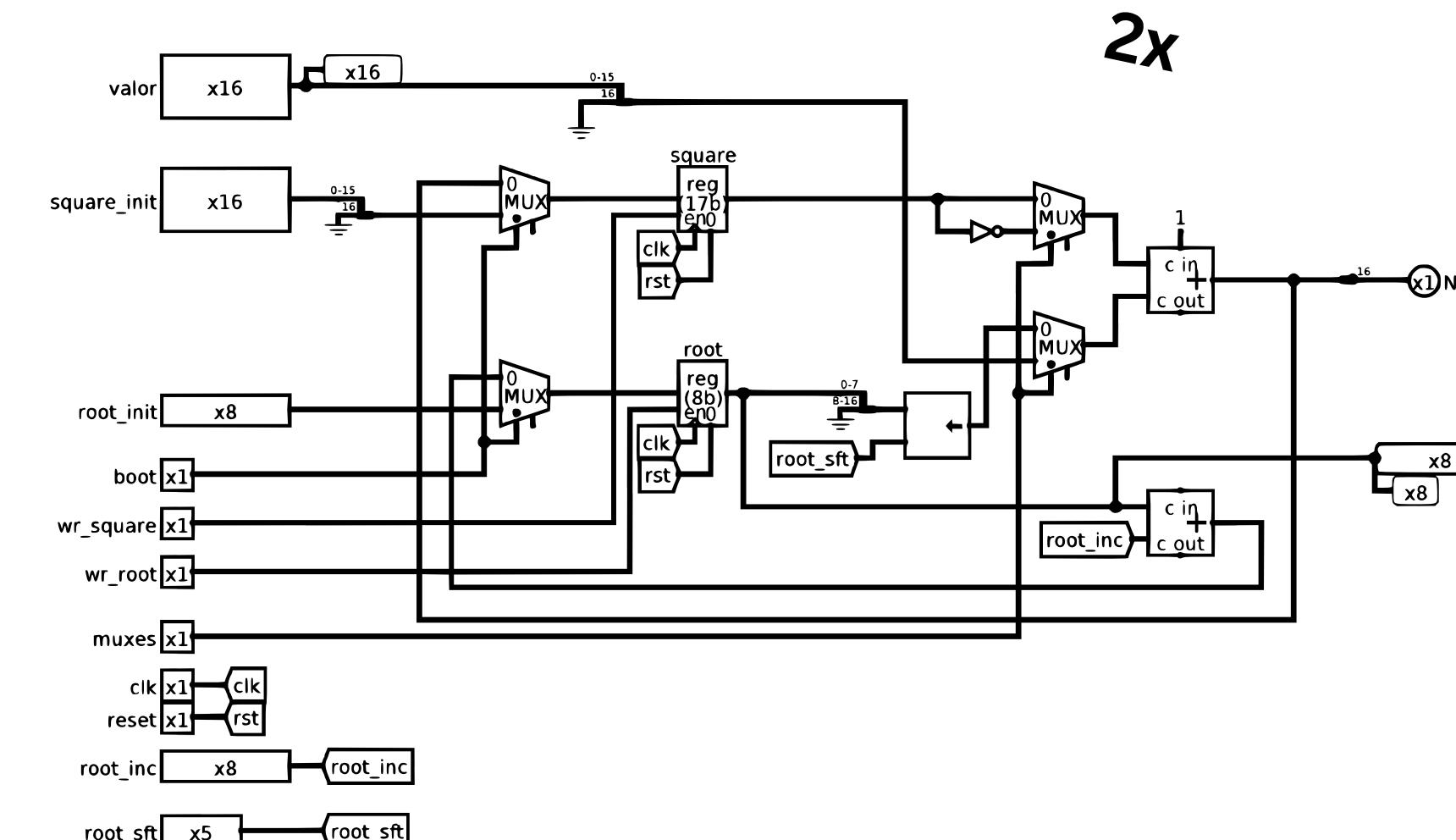
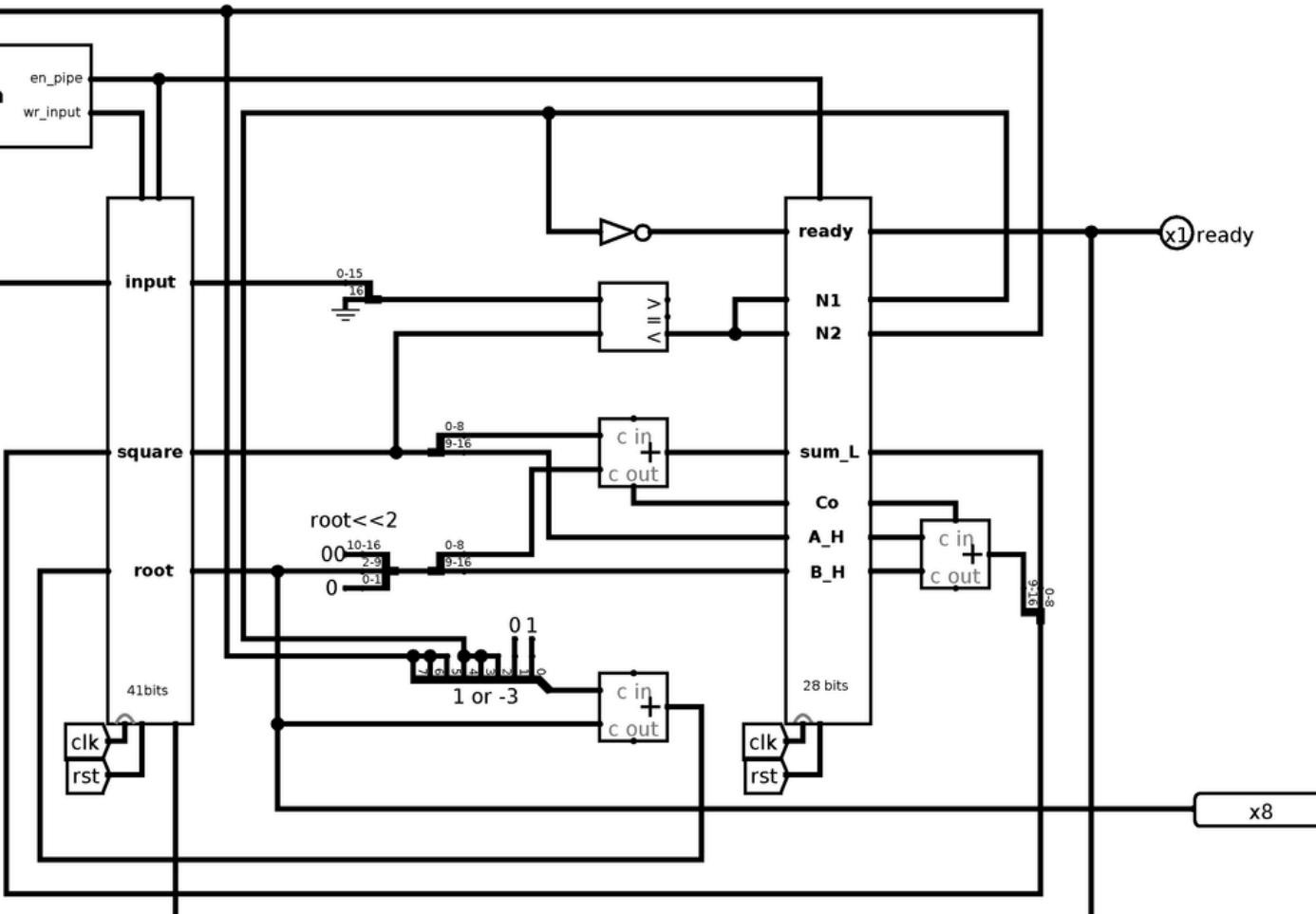
Versão 1



DATA PATH



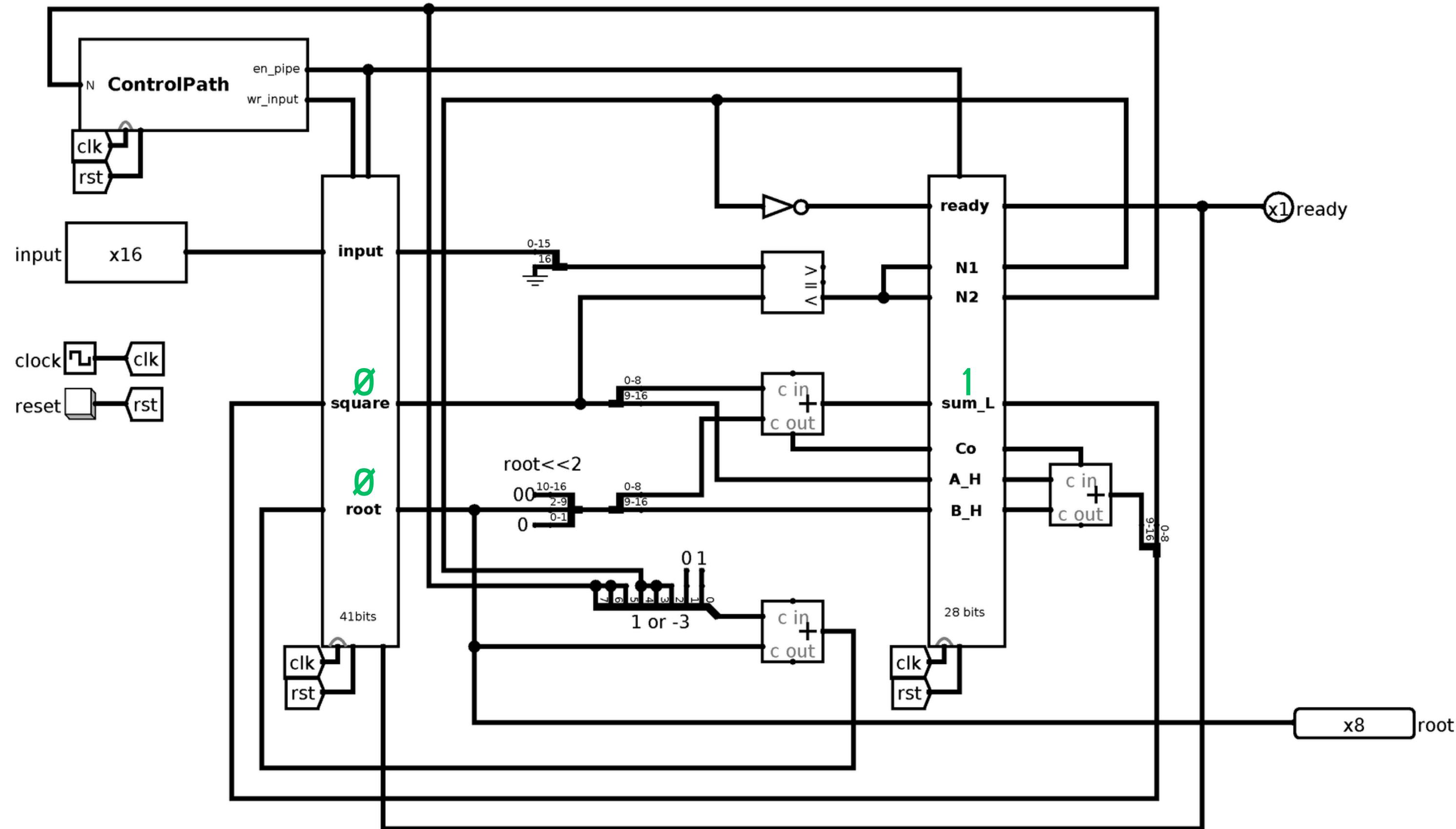
DATA PATH



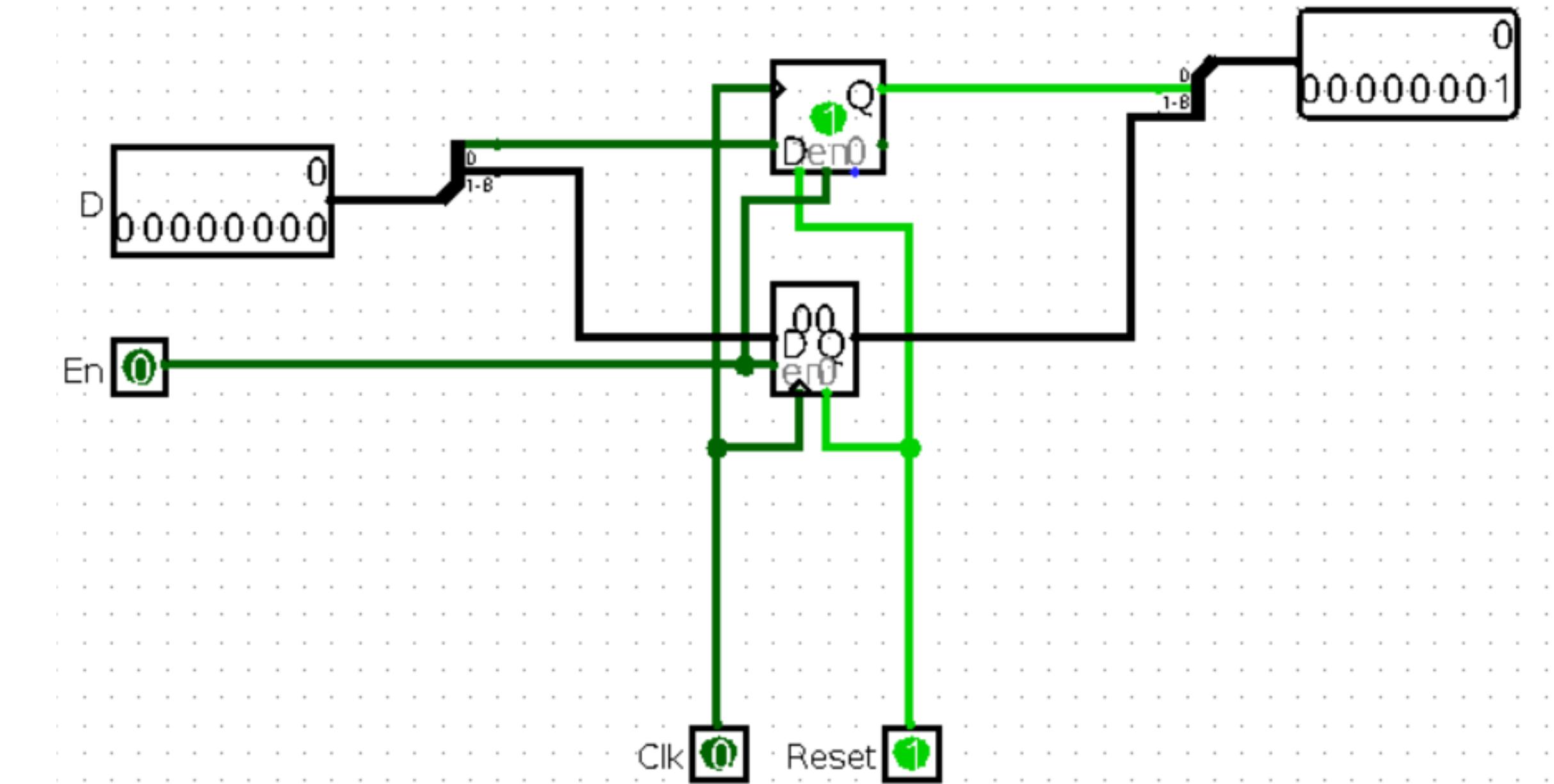
• Hardware Utilizado

VERSÃO	FLIP-FLOPS	SOMADORES	MUXES	COMPARADORES
ORIGINAL	66	2x17 bits 2x8 bits	9	0
2-STAGE PIPELINE	70	1x9 bits 2x8 bits	0	1
4 STAGE PIPELINE	132	1x5 bits 5x4 bits	0	1

DATA PATH

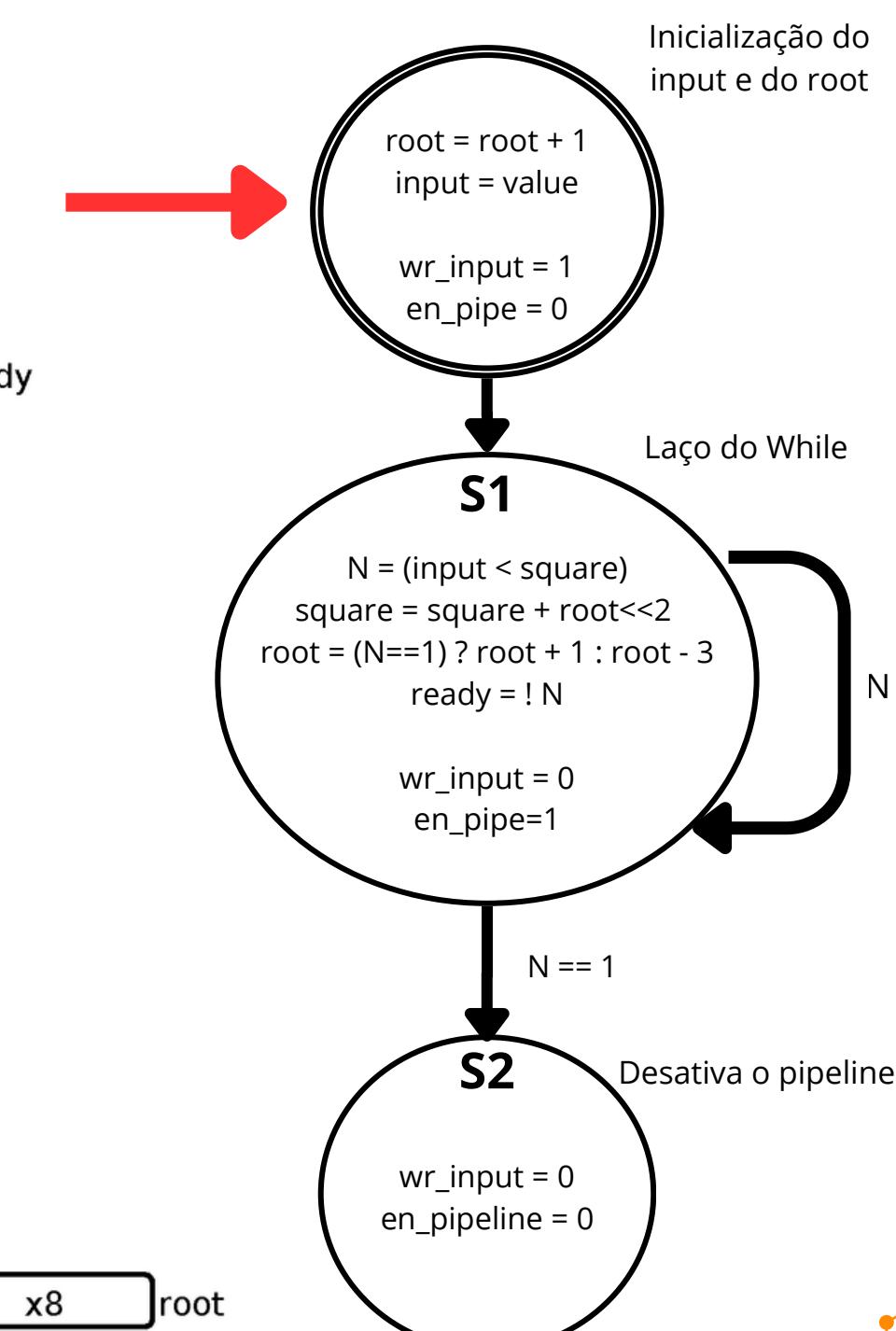
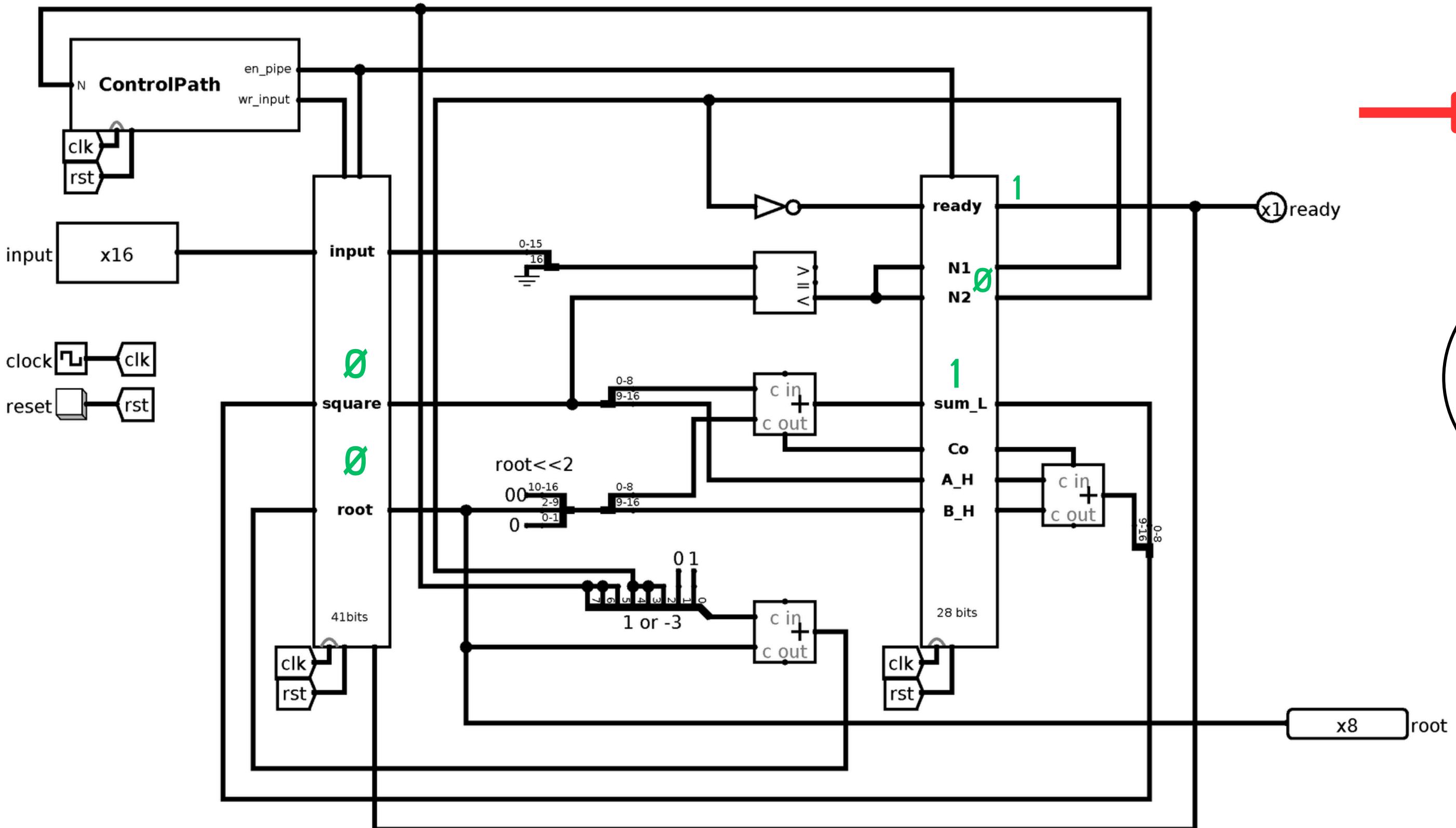


DATA PATH



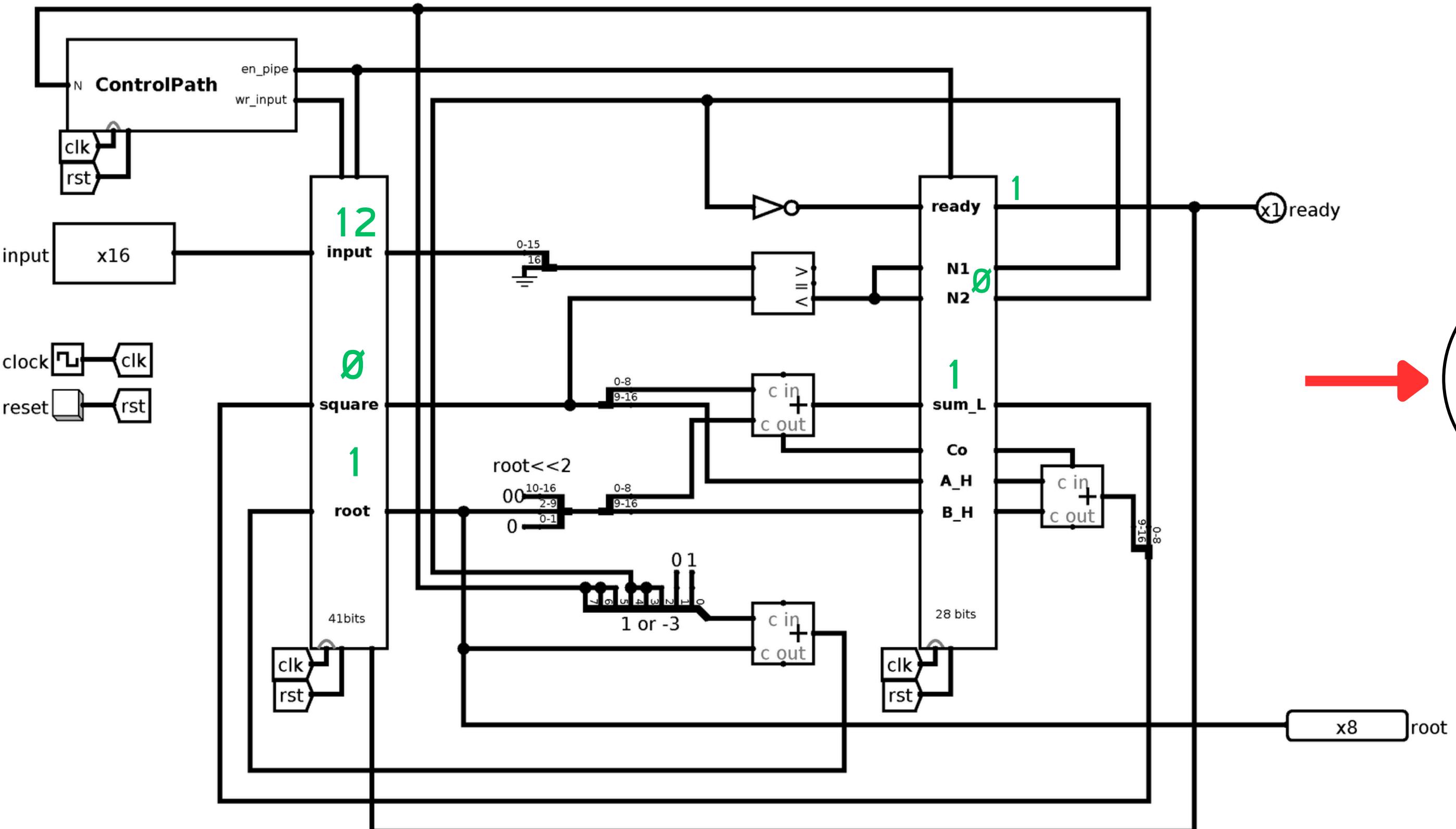
DATA PATH

CICLO 1



DATA PATH

CICLO 2



Inicialização do input e do root

$\text{root} = \text{root} + 1$
 $\text{input} = \text{value}$
 $\text{wr_input} = 1$
 $\text{en_pipe} = 0$

Laço do While

S1
 $N = (\text{input} < \text{square})$
 $\text{square} = \text{square} + \text{root} \ll 2$
 $\text{root} = (N == 1) ? \text{root} + 1 : \text{root} - 3$
 $\text{ready} = ! N$
 $\text{wr_input} = 0$
 $\text{en_pipe} = 1$

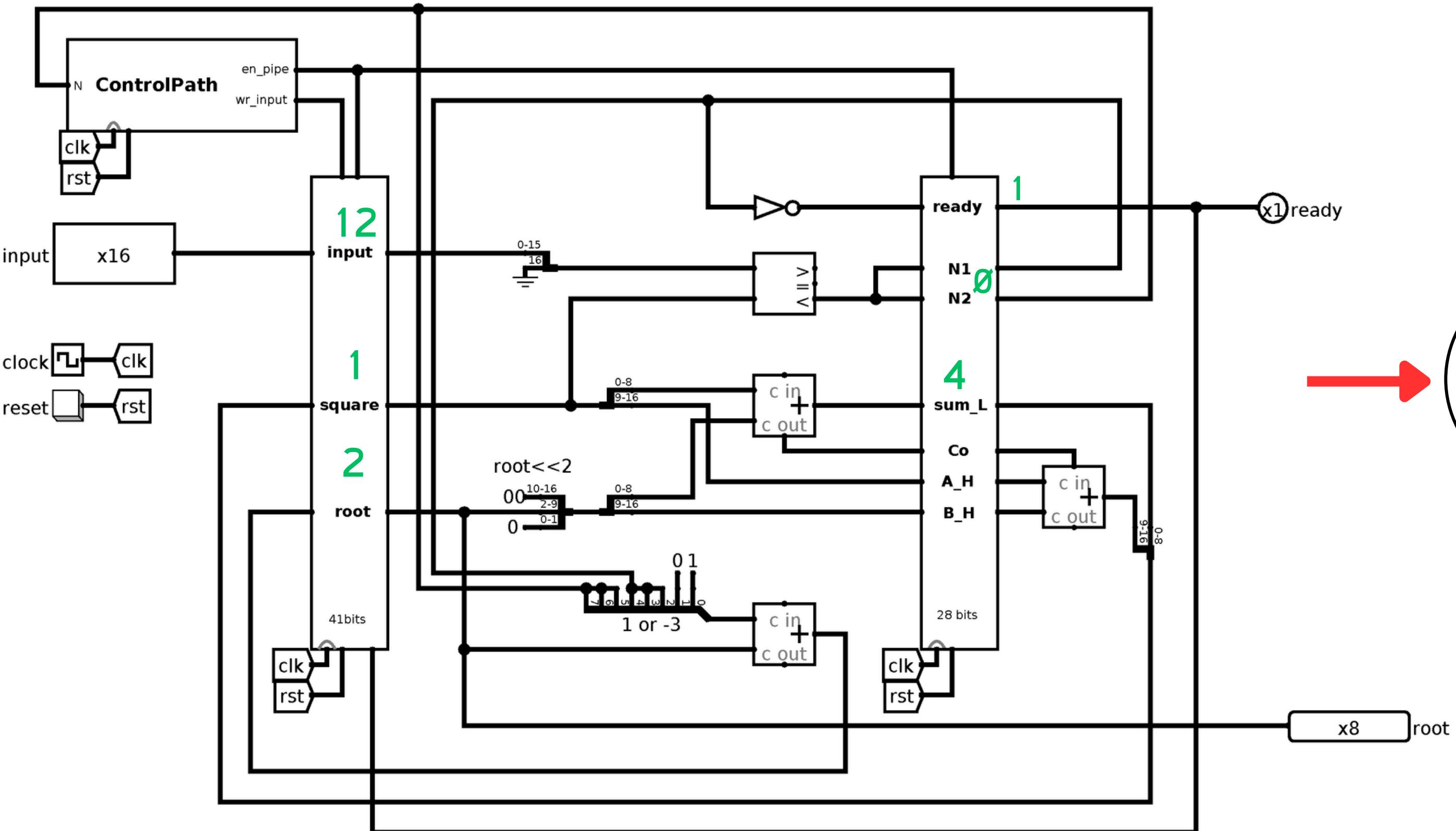
$N == 0$

Desativa o pipeline

$\text{wr_input} = 0$
 $\text{en_pipe} = 0$

DATA PATH

CICLO 3



Inicialização do input e do root

$\text{root} = \text{root} + 1$
 $\text{input} = \text{value}$
 $\text{wr_input} = 1$
 $\text{en_pipe} = 0$

Laço do While

S1
 $\text{N} = (\text{input} < \text{square})$
 $\text{square} = \text{square} + \text{root} \ll 2$
 $\text{root} = (\text{N} == 1) ? \text{root} + 1 : \text{root} - 3$
 $\text{ready} = ! \text{N}$
 $\text{wr_input} = 0$
 $\text{en_pipe} = 1$

$\text{N} == 0$

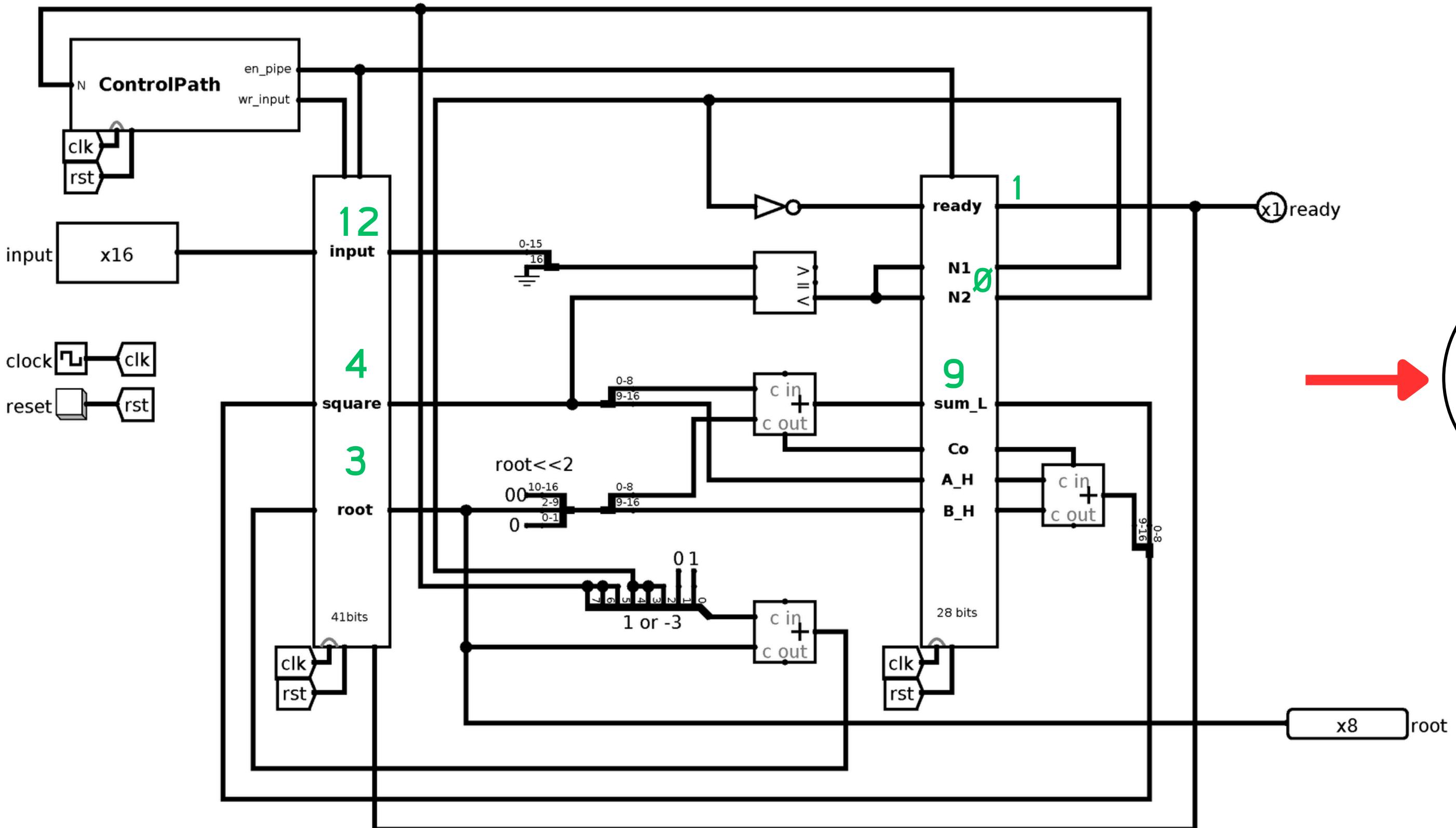
Desativa o pipeline

$\text{wr_input} = 0$
 $\text{en_pipe} = 0$

15

DATA PATH

CICLO 4



Inicialização do input e do root

$\text{root} = \text{root} + 1$
 $\text{input} = \text{value}$
 $\text{wr_input} = 1$
 $\text{en_pipe} = 0$

Laço do While

S1
 $N = (\text{input} < \text{square})$
 $\text{square} = \text{square} + \text{root} << 2$
 $\text{root} = (N == 1) ? \text{root} + 1 : \text{root} - 3$
 $\text{ready} = !N$
 $\text{wr_input} = 0$
 $\text{en_pipe} = 1$

$N == 0$

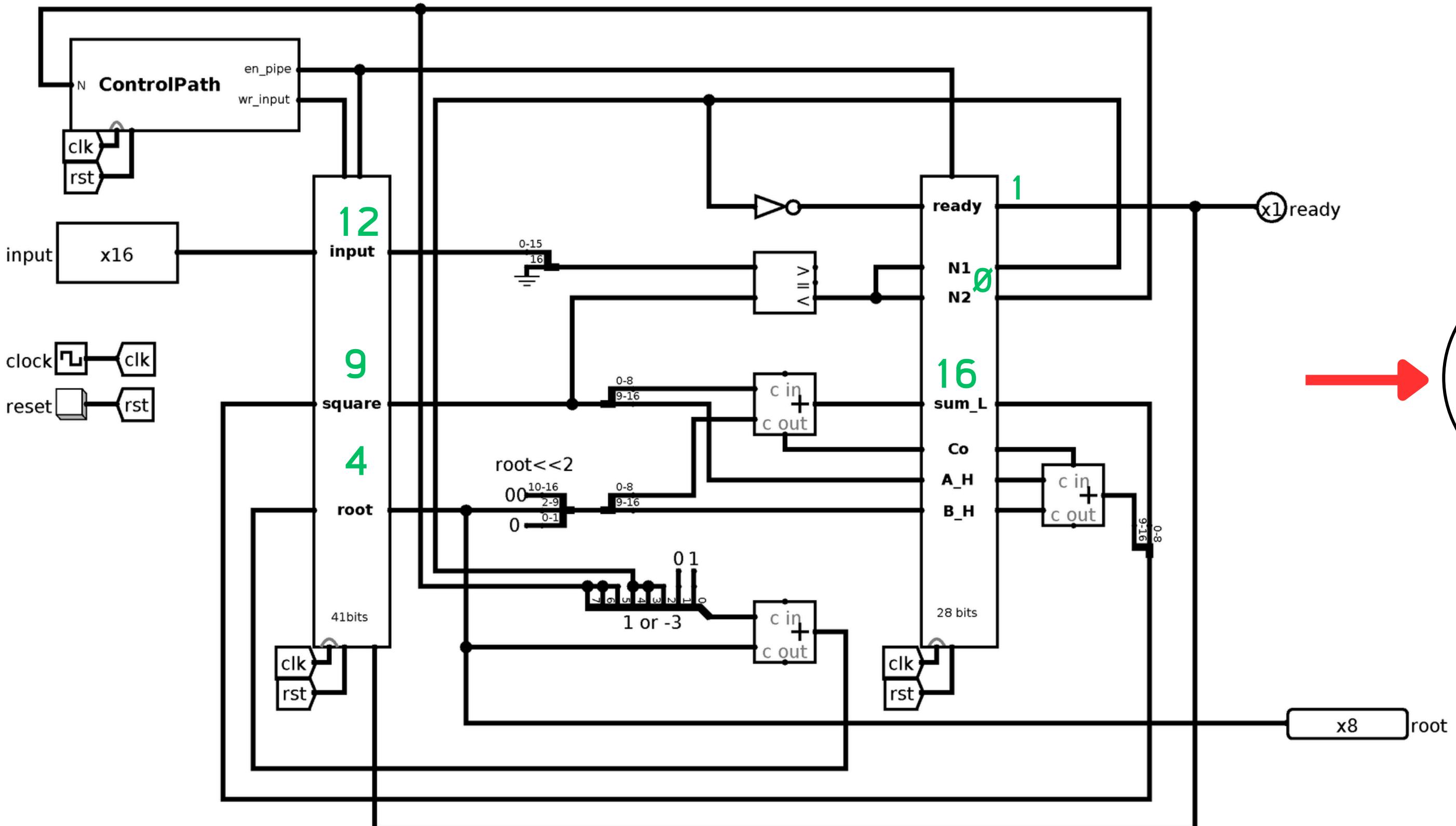
Desativa o pipeline

$\text{wr_input} = 0$
 $\text{en_pipe} = 0$

16

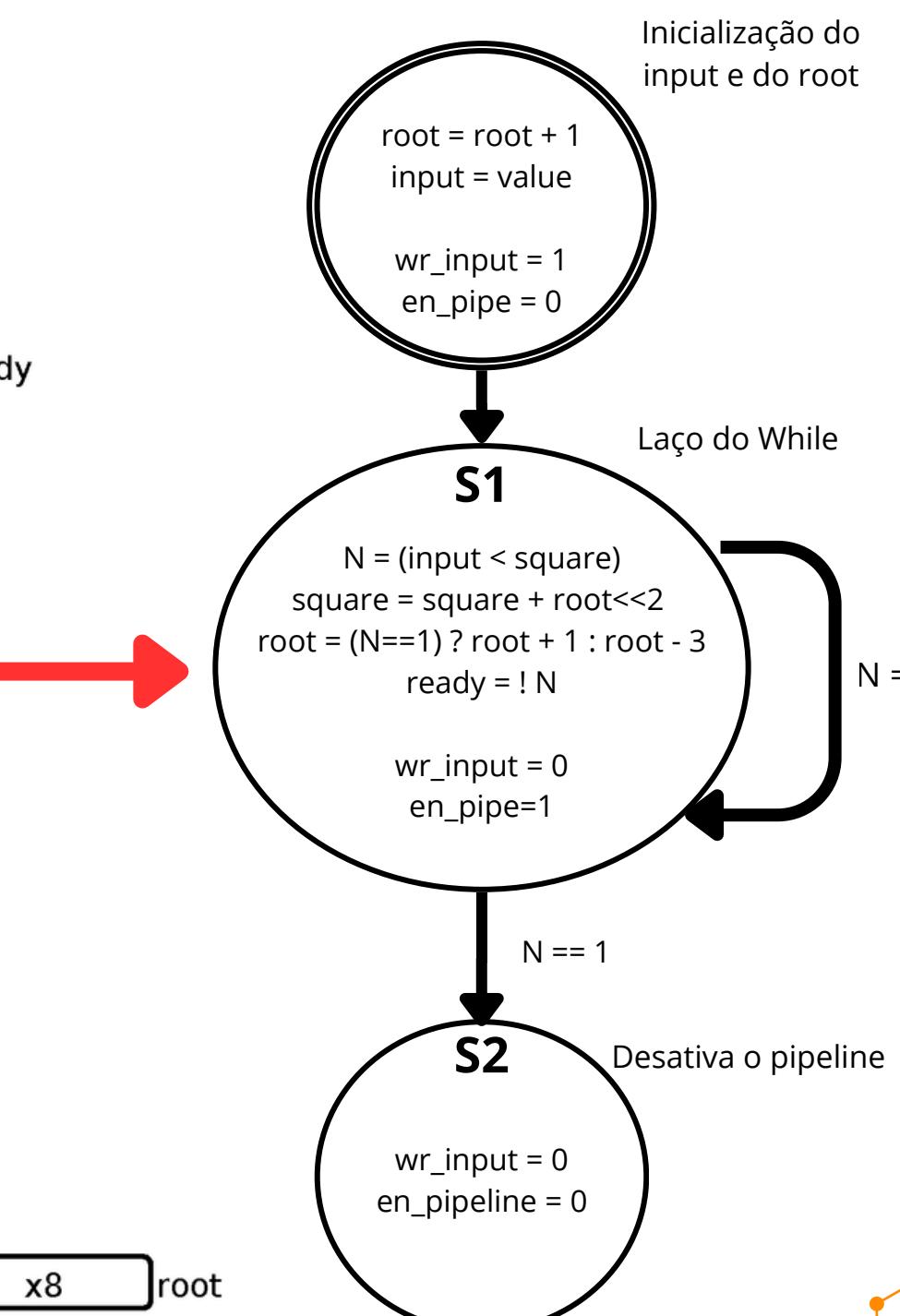
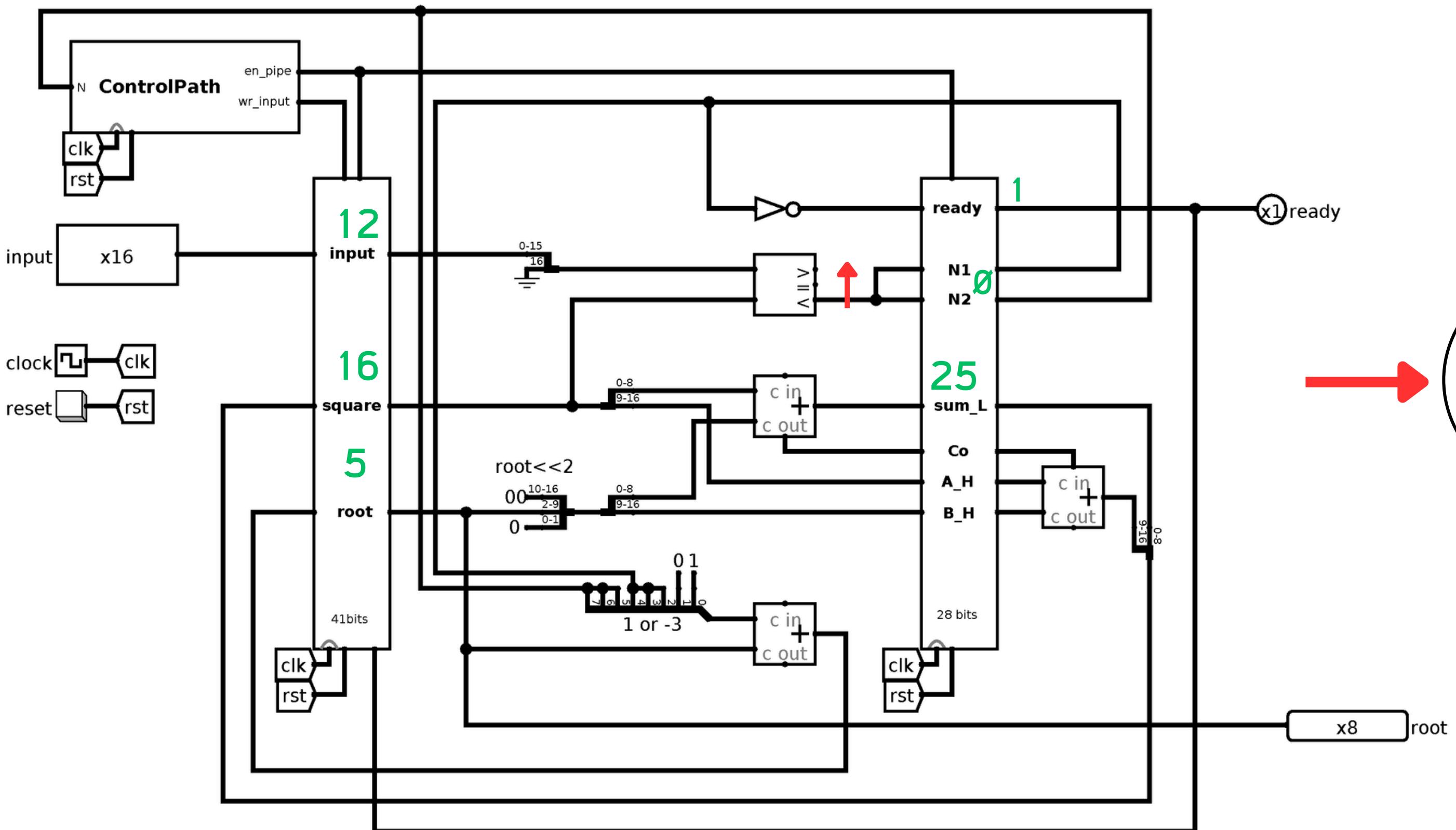
DATA PATH

CICLO 5



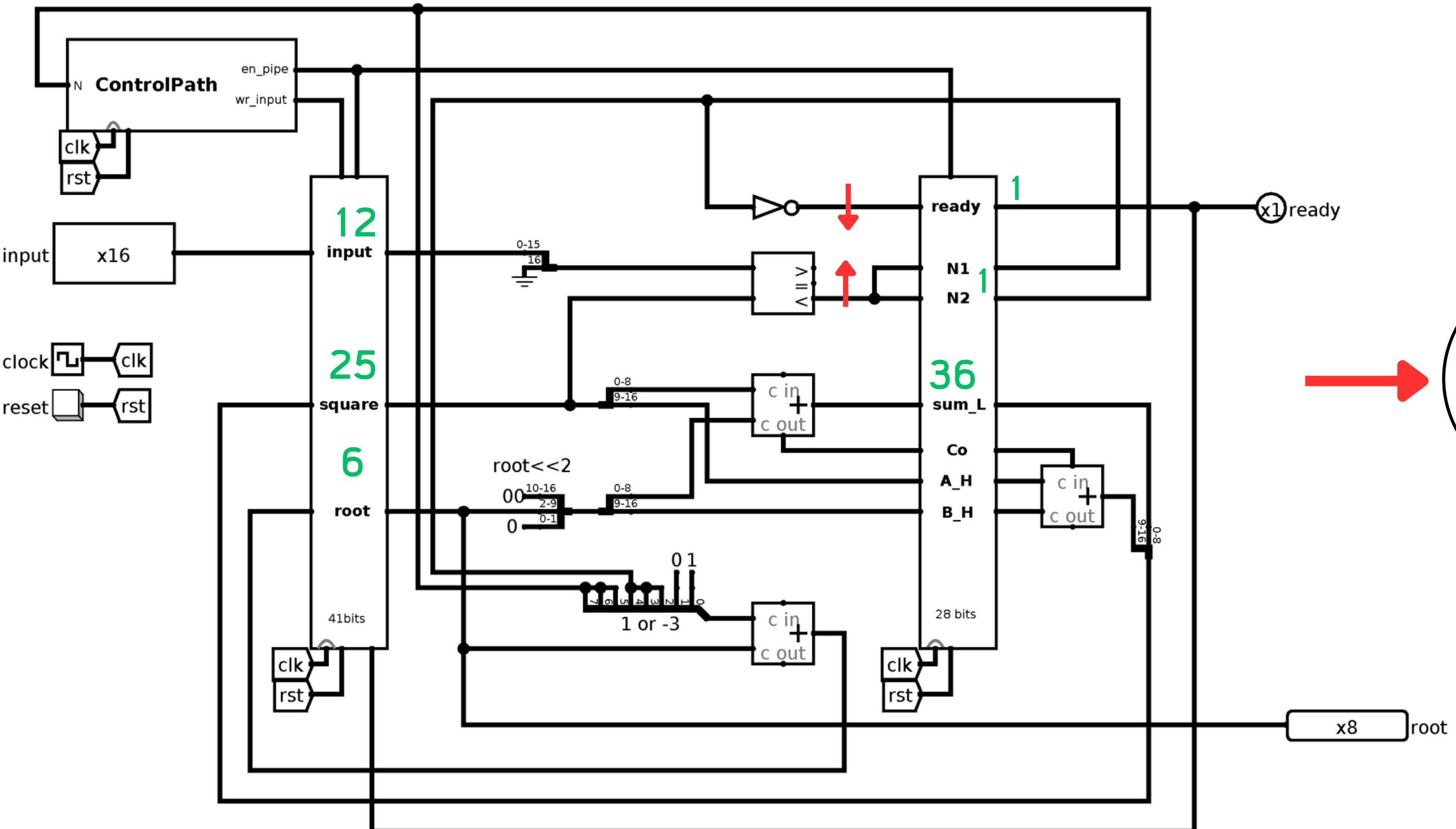
DATA PATH

CICLO 6



DATA PATH

CICLO 7



Inicialização do input e do root

root = root + 1
input = value
wr_input = 1
en_pipe = 0

Laço do While

S1
N = (input < square)
square = square + root<<2
root = (N==1) ? root + 1 : root - 3
ready = ! N
wr_input = 0
en_pipe=1

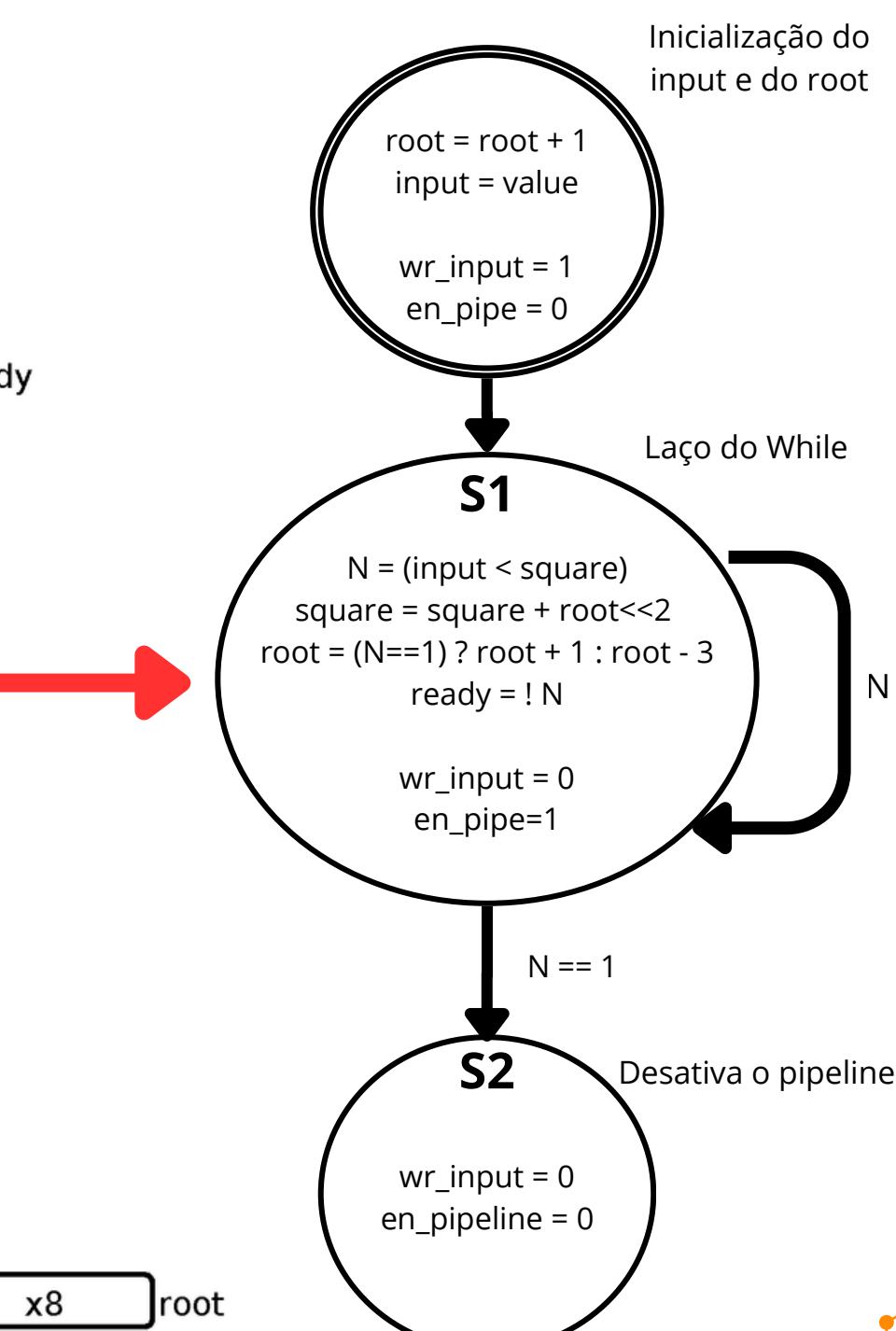
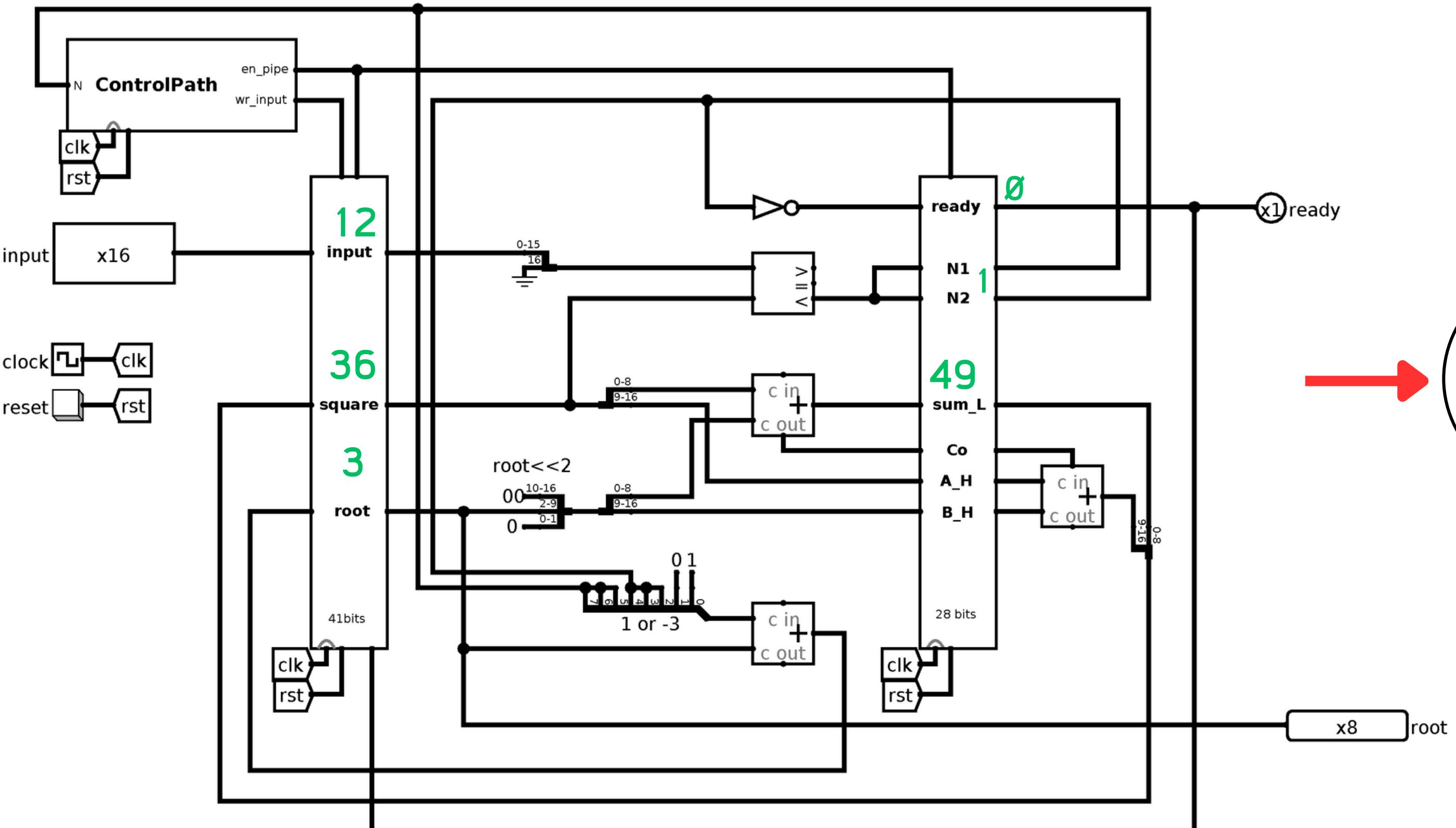
N == 0

Desativa o pipeline

wr_input = 0
en_pipeline = 0

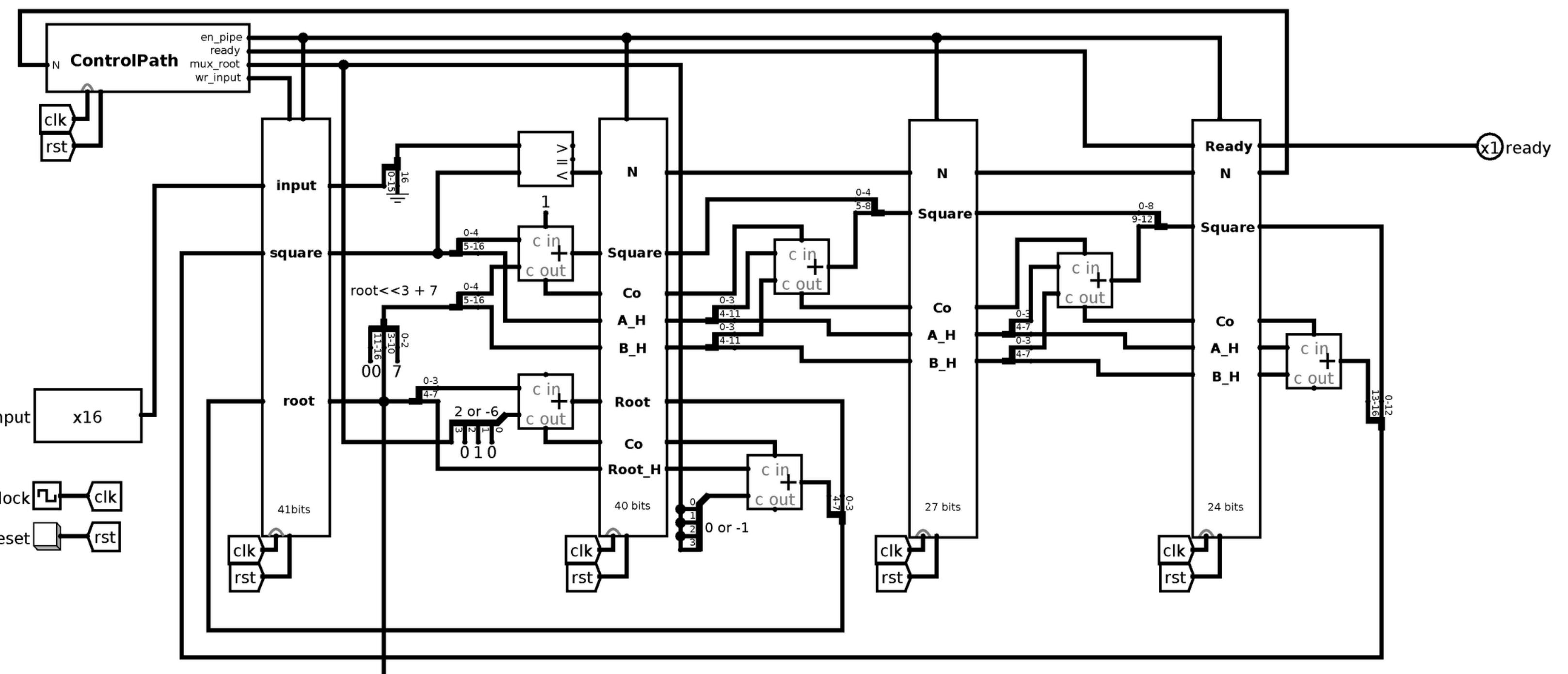
DATA PATH

CIC



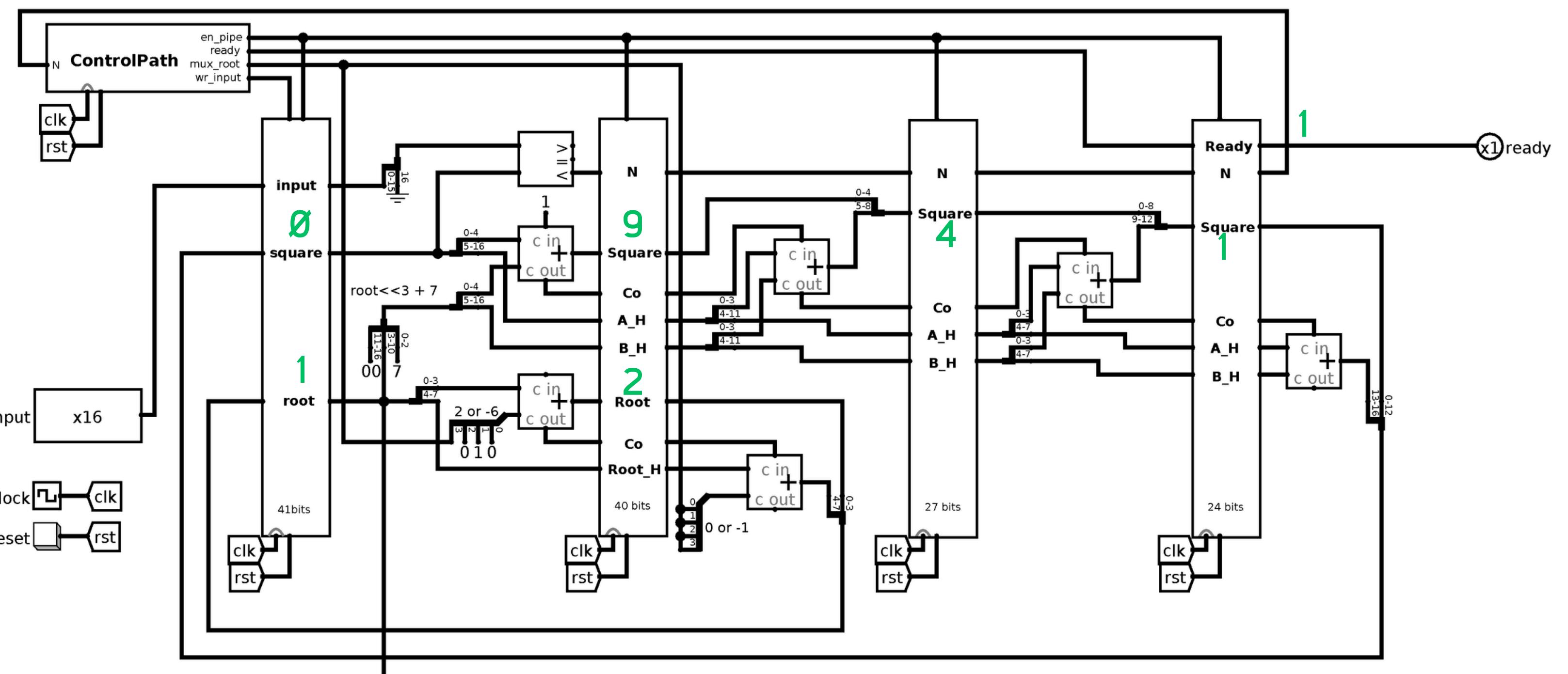
DATA PATH

PIPELINE COM QUATRO ESTÁGIOS



DATA PATH

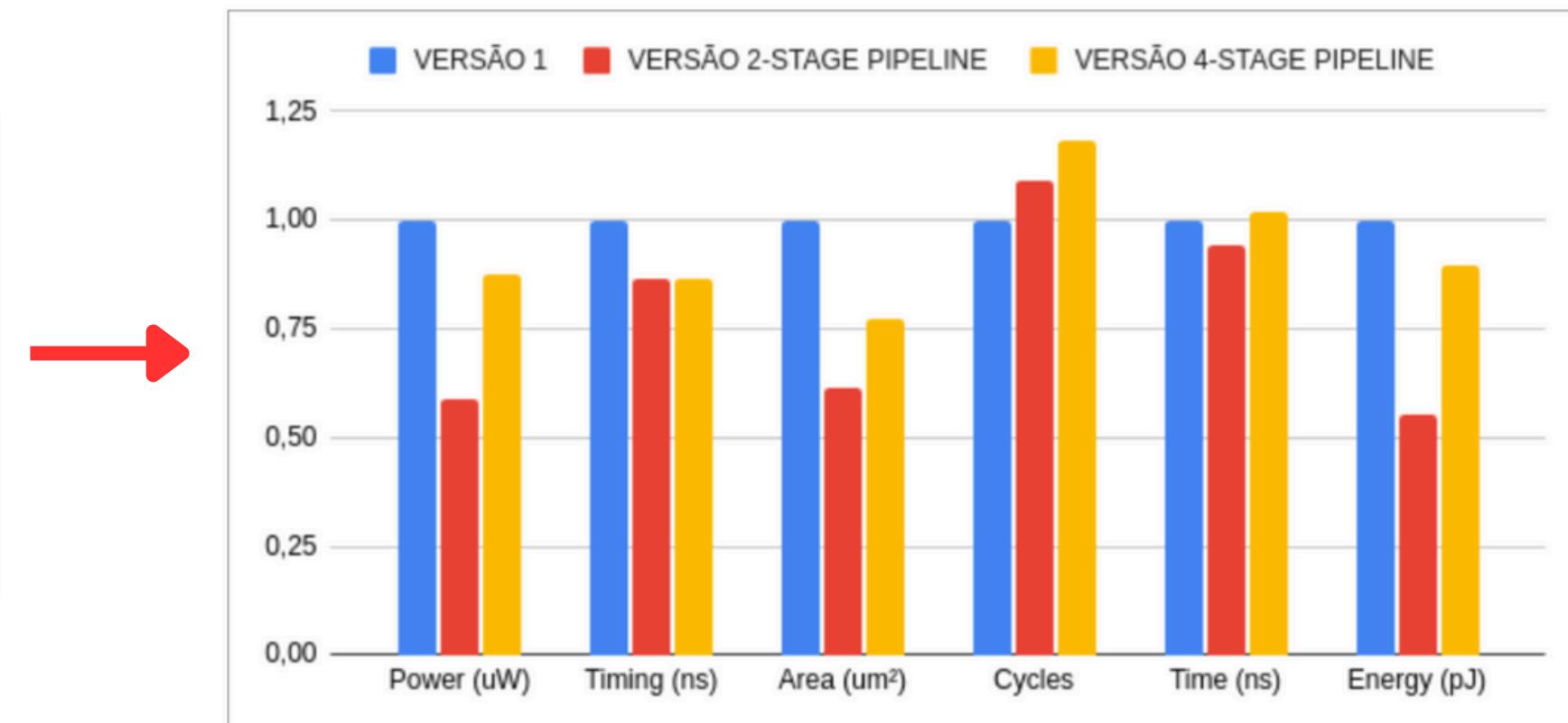
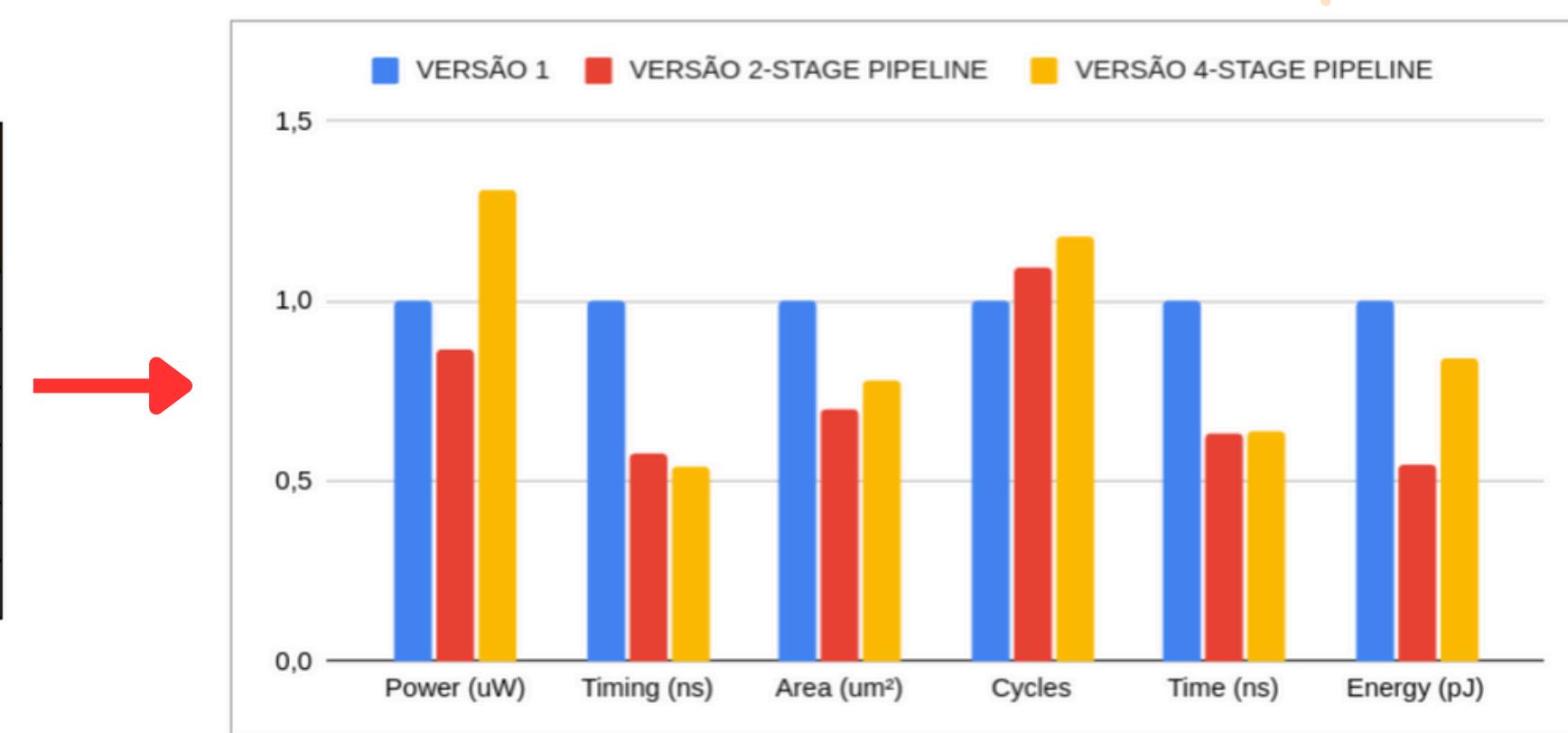
PIPELINE COM QUATRO ESTÁGIOS



RESULTADOS

WORST	VERSÃO 1	VERSÃO 2-STAGE PIPELINE	VERSÃO 4-STAGE PIPELINE
Power (uW)	493,969	427,74	- 13,41 %
Timing (ns)	2,38	1,38	- 42,02 %
Area (μm^2)	2504,094	1749,832	- 30,12 %
Cycles	22	24	+ 2
Time (ns)	52,360	33,120	- 36,75 %
Energy (pJ)	25,86	14,17	- 45,23 %

BEST	VERSÃO 1	VERSÃO 2-STAGE PIPELINE	VERSÃO 4-STAGE PIPELINE
Power (uW)	4.458,020	2.625,45	- 41,11 %
Timing (ns)	0,51	0,44	- 13,73 %
Area (μm^2)	1988,091	1221,624	- 38,55 %
Cycles	22	24	+ 2
Time (ns)	11,220	10,560	- 5,88 %
Energy (pJ)	50,02	27,72	- 44,57 %



COMPARAÇÃO

WORST	JOÃO - 351MHz	THIS WORK - 727MHz	THIS WORK - 351MHz
Power (uW)	134,000	427,74	+ 219,21 %
Timing (ns)	2,85	1,38	- 51,58 %
Area (um ²)	1082,284	1749,832	+ 51,58 %
Cycles	21	24	+ 3
Time (ns)	59,850	33,120	- 44,66 %
Energy (pJ)	8,02	14,17	+ 76,65 %

