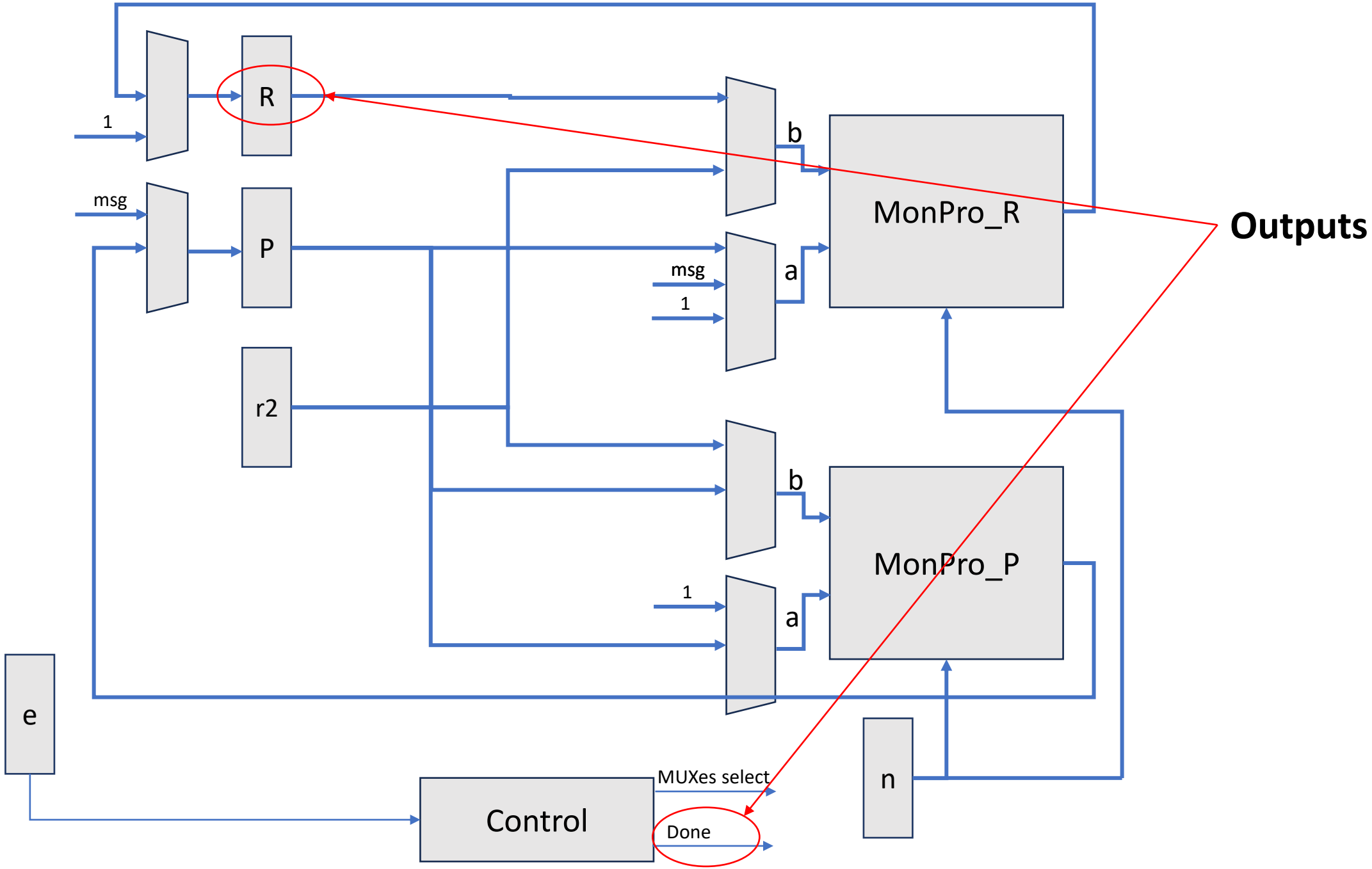
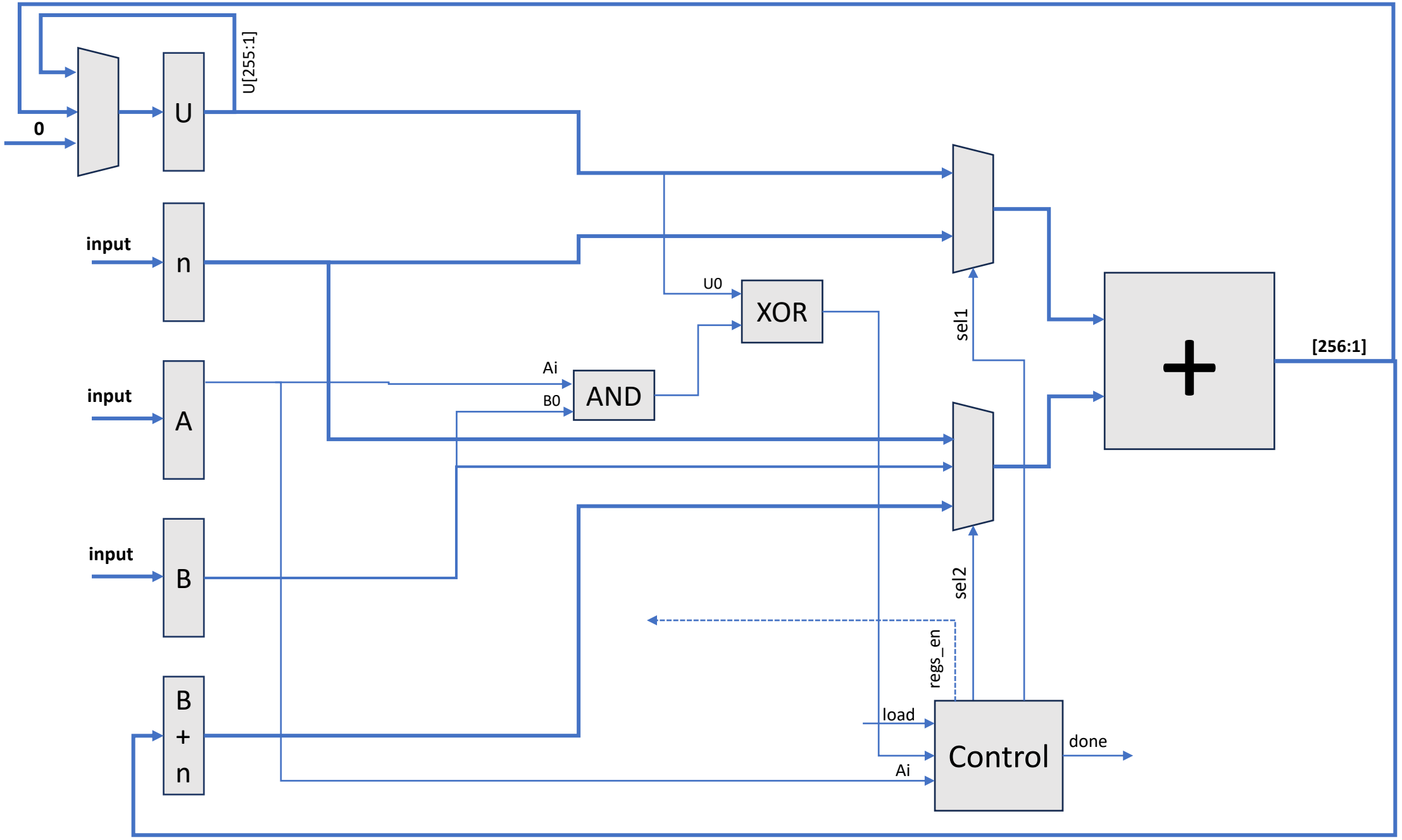
# The algorithm, and why

- Exponentiation algorithm: Binary Right to Left
  - →Simple square/multiply loop.
  - →Product and Square are independent operations, can be done in parallel.
- Multiplication algorithm: Montgomery
  - →Instead of division by n, a division by power of 2 is done, which is a shift operation and is much more efficient in hardware.
  - →After a preprocessing step, only one addition operation is needed inside the loop that implements the algorithm.

# The microarchitecture: MonExp

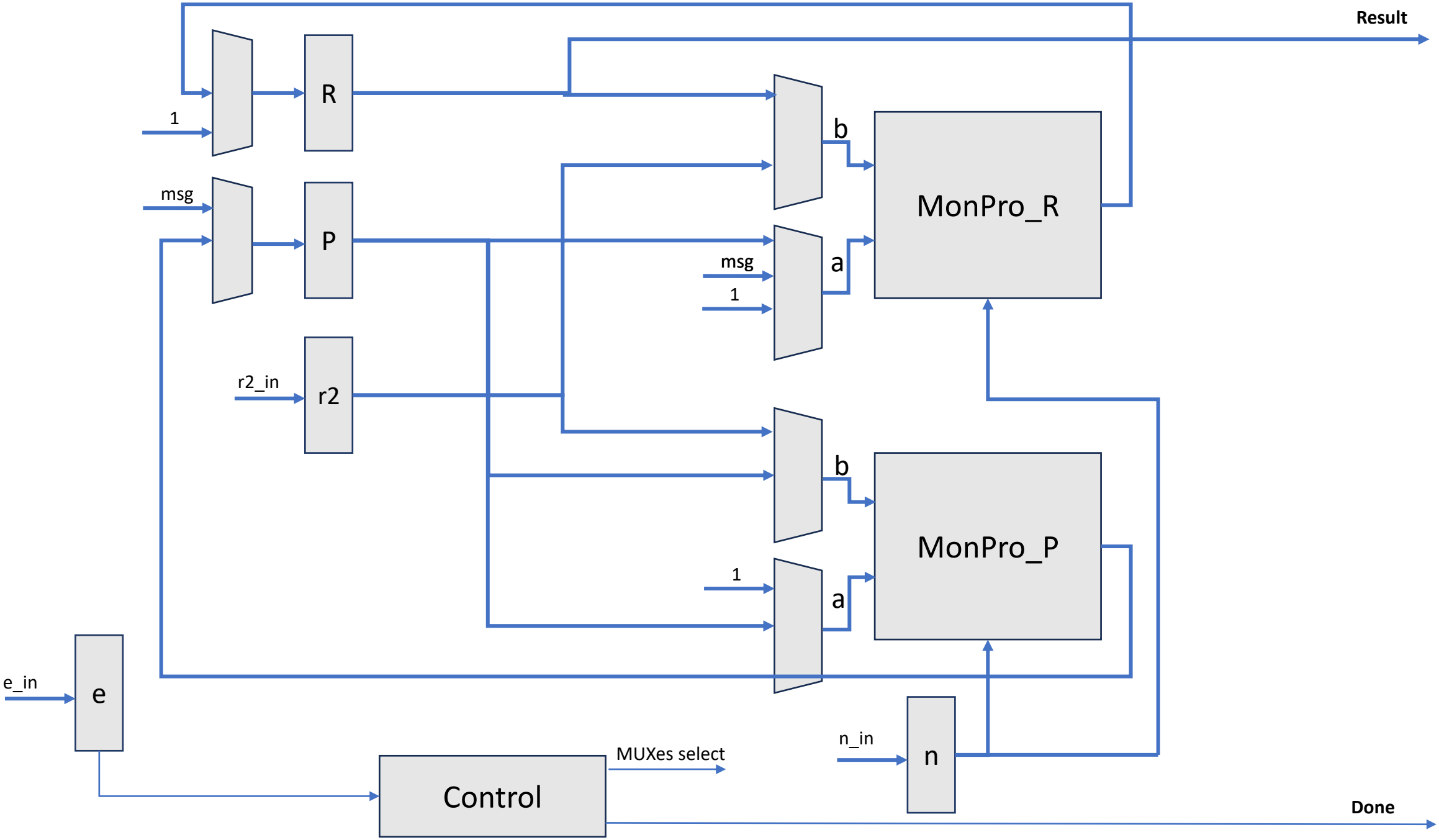

R
1
msg
P
r2
b
a
msg
1
MonPro_R
b
a
1
MonPro_P
Outputs
e
Control
MUXes select
Done
n

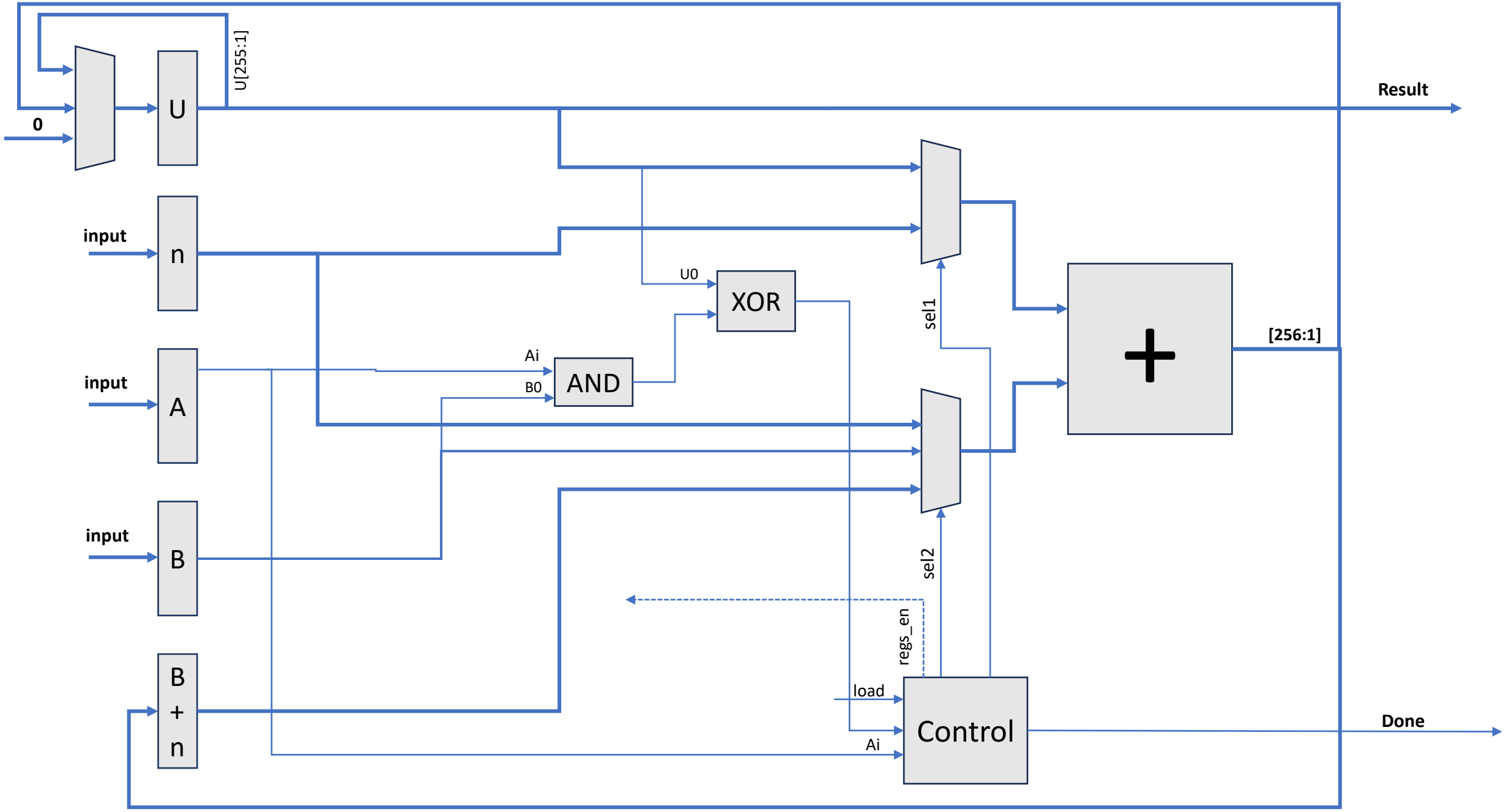
# The microarchitecture: MonPro


0
U
U[255:1]
input
n
input
A
input
B
B
+
n
U0
XOR
Ai
B0
AND
sel1
sel2
+
[256:1]
regs_en
load
Ai
Control
done

# Performance estimation

- Clocks cycles per operation: 258*258 = 66564 cycles
- Estimated frequency: 120MHz
  - Generated from synthesizing MonPro, which has a 256-bit adder, which is our bottleneck.
  - Synthesis strategy: Performance Optimized
- Throughput: 1772 exponentiations per second


Result
R
1
msg
P
r2_in
r2
b
a
msg
1
MonPro_R
b
a
1
MonPro_P
e_in
e
n_in
n
MUXes select
Control
Done


0
U
U[255:1]
Result
input
n
U0
XOR
sel1
input
A
Ai
B0
AND
[256:1]
input
B
sel2
regs_en
B
+
n
load
Control
Done
Ai