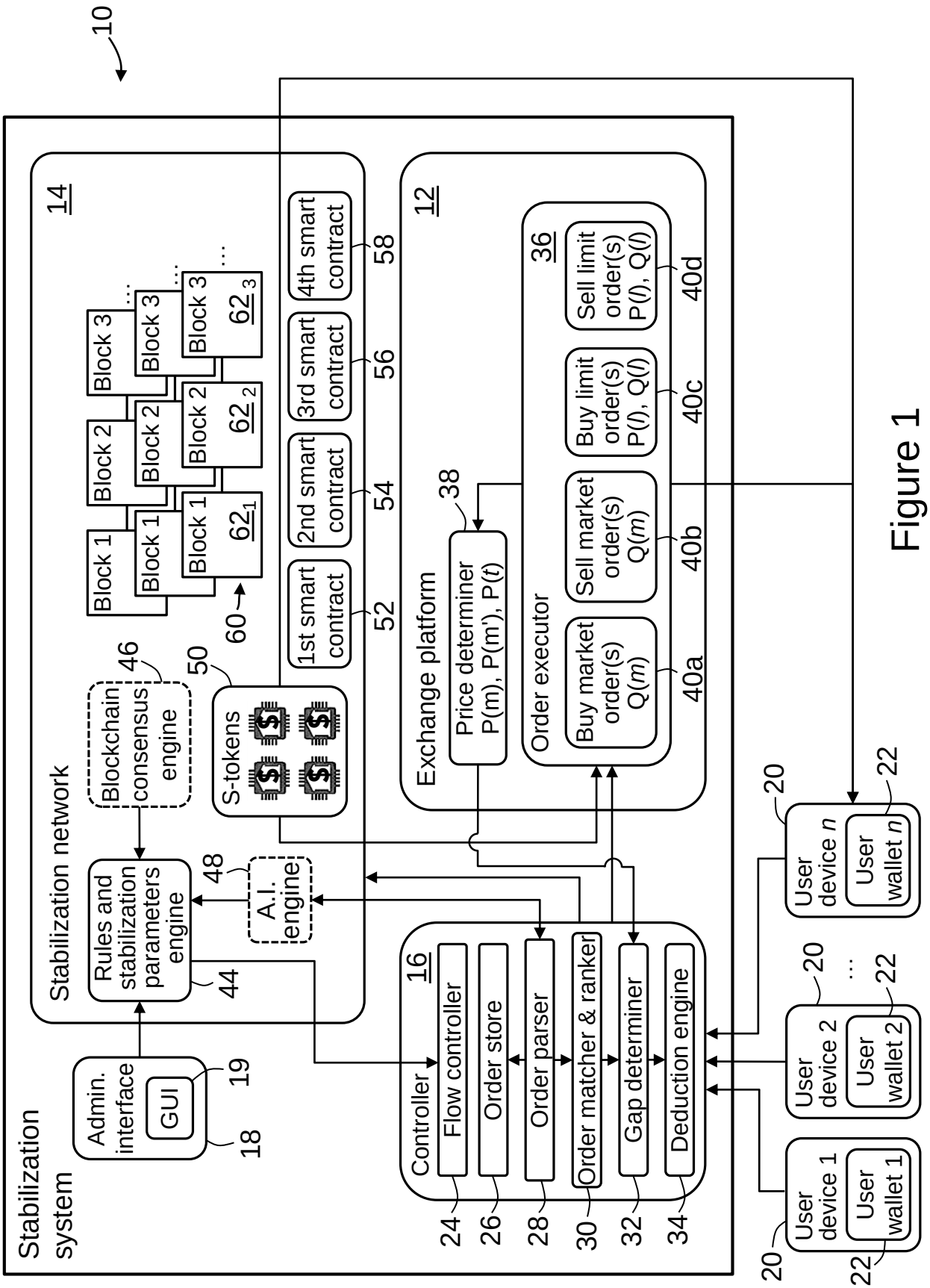

10
Stabilization system
Admin. interface
GUI
19
18
Stabilization network
14
Rules and stabilization parameters engine
44
48
A.I. engine
Blockchain consensus engine
46
50
S-tokens
60
Block 1
Block 2
Block 3
62 1
62 2
62 3
1st smart contract
52
2nd smart contract
54
3rd smart contract
56
4th smart contract
58
Exchange platform
12
Price determiner P(m), P(m'), P(t)
38
Order executor
36
Buy market order(s) Q(m)
40a
Sell market order(s) Q(m)
40b
Buy limit order(s) P(l), Q(l)
40c
Sell limit order(s) P(l), Q(l)
40d
Controller
16
Flow controller
Order store
Order parser
Order matcher & ranker
Gap determiner
Deduction engine
24
26
28
30
32
34
User device 1
User wallet 1
User device 2
User wallet 2
User device n
User wallet n
20
22

Figure 1

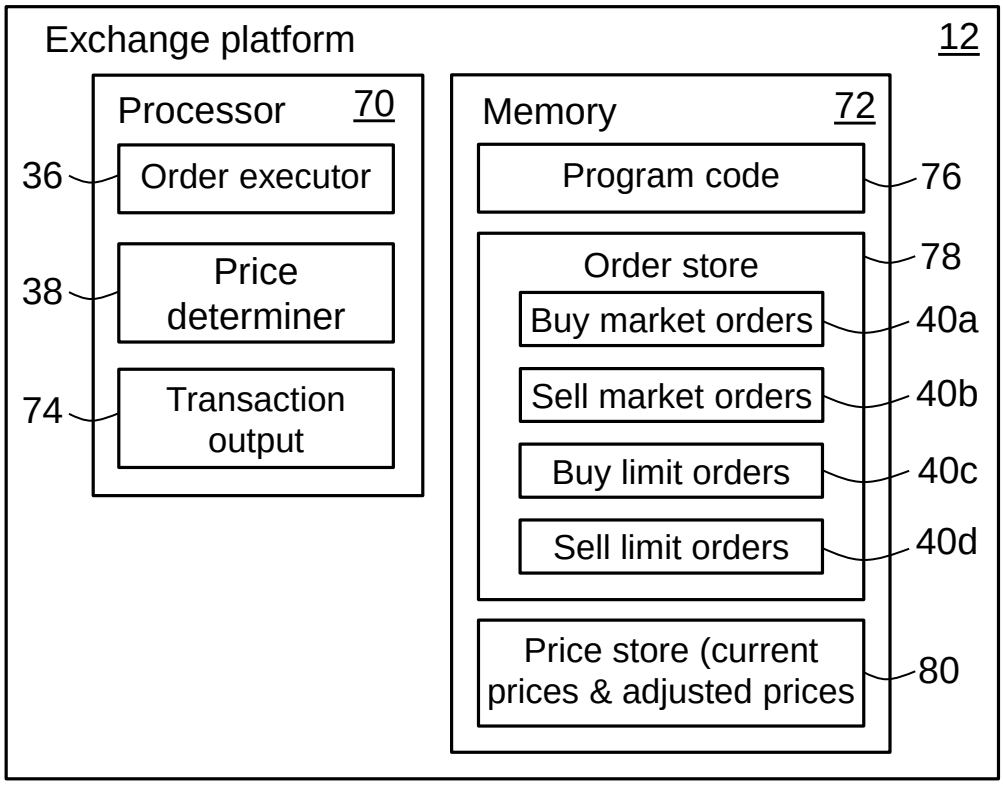


Figure 2A

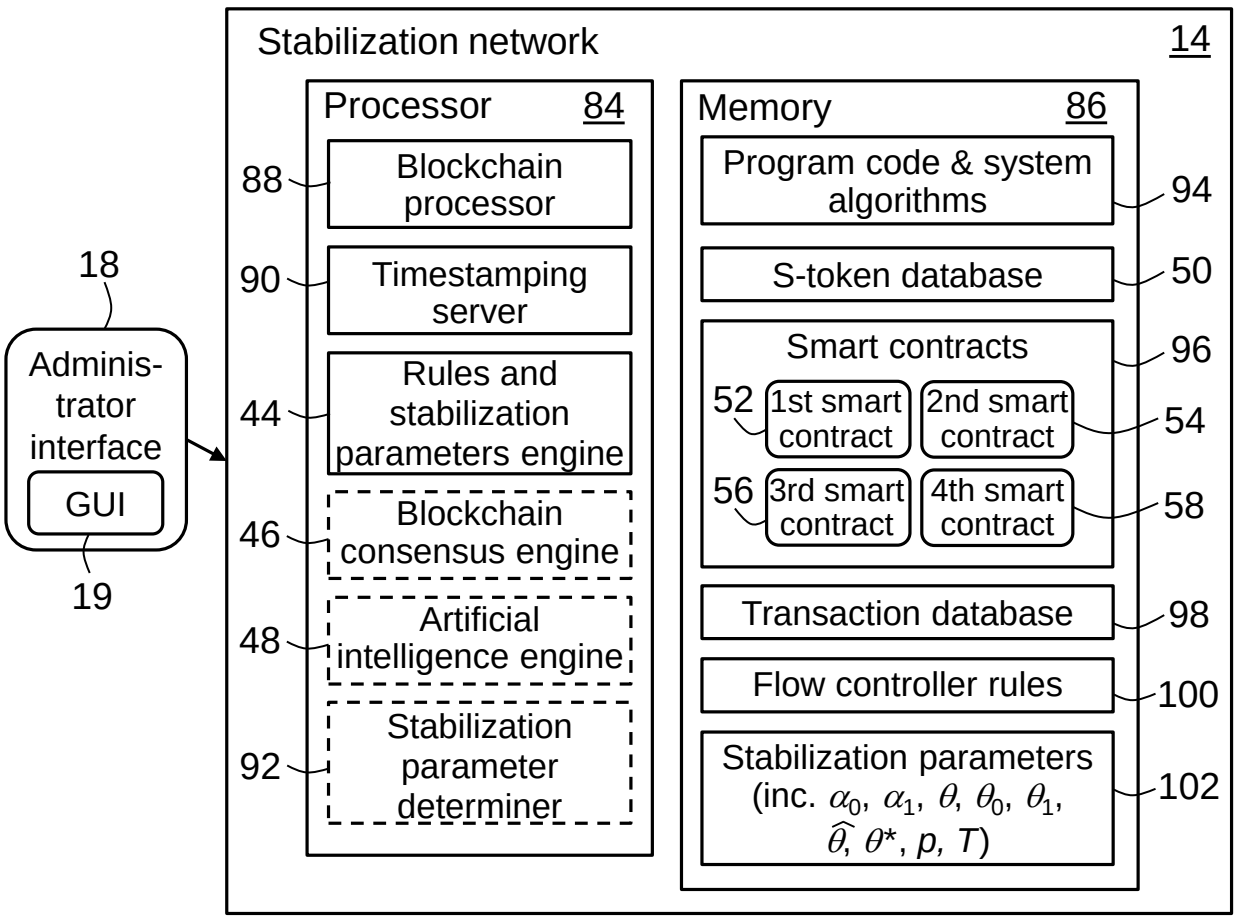


Figure 2B

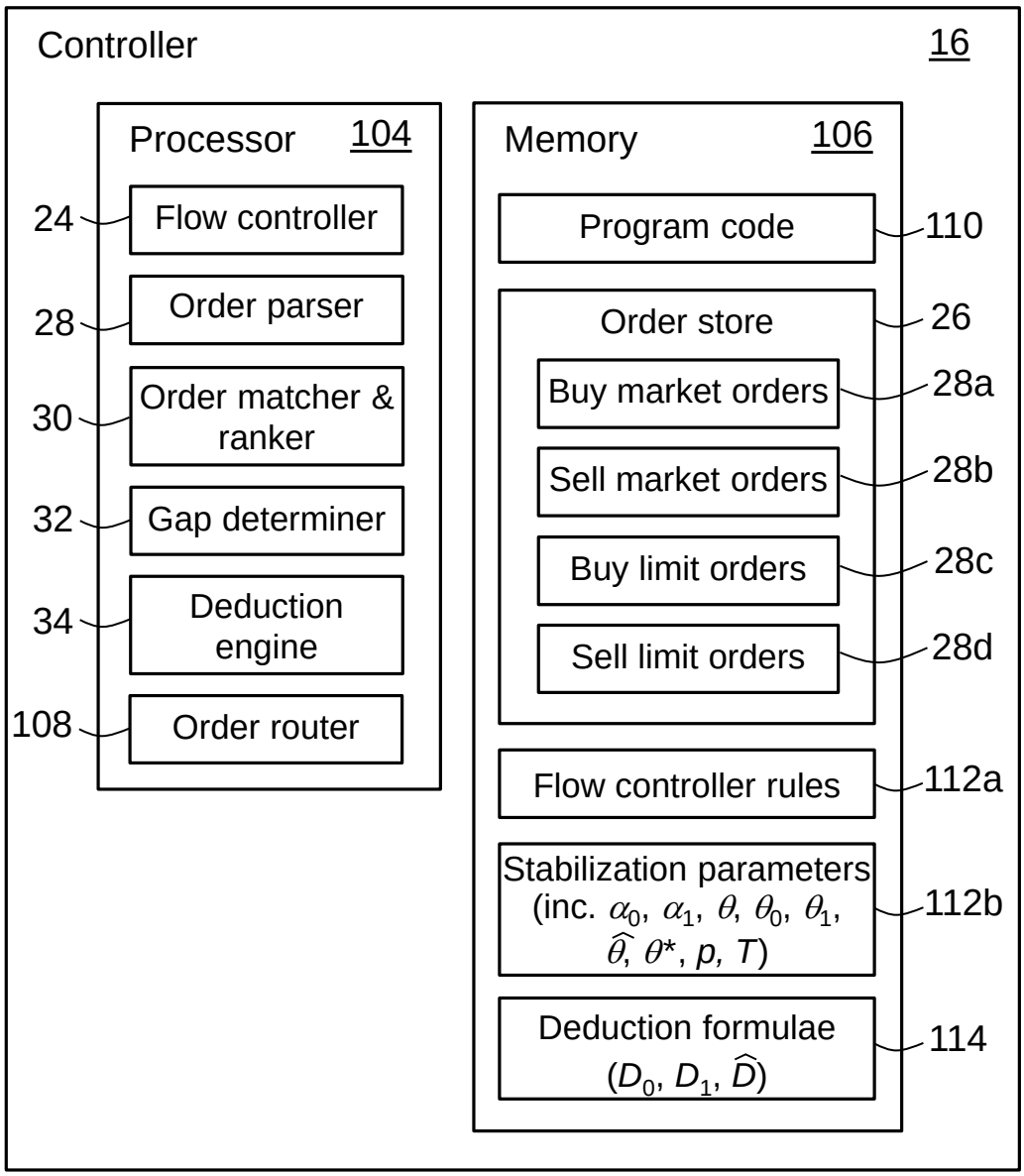

Controller
16
Processor
104
24
Flow controller
28
Order parser
30
Order matcher & ranker
32
Gap determiner
34
Deduction engine
108
Order router
Memory
106
Program code
110
Order store
26
Buy market orders
28a
Sell market orders
28b
Buy limit orders
28c
Sell limit orders
28d
Flow controller rules
112a
Stabilization parameters (inc. α_0, α_1, θ, θ_0, θ_1, $\widehat{\theta}$, θ^*, p, T)
112b
Deduction formulae (D_0, D_1, $\widehat{D}$)
114

Figure 2C

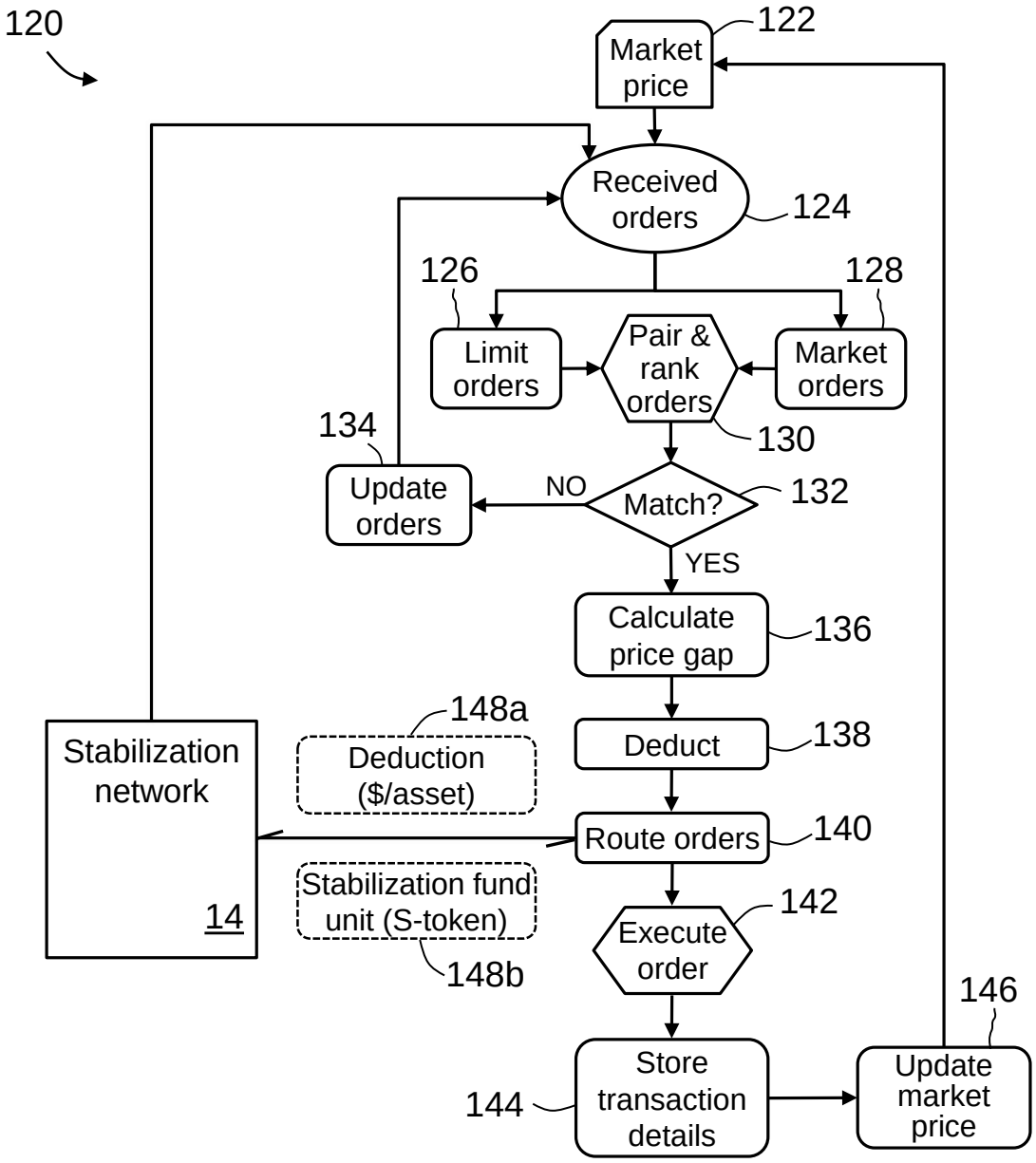

120
Market price
122
Received orders
124
126
128
Limit orders
Pair & rank orders
Market orders
130
134
Update orders
NO
Match?
132
YES
Calculate price gap
136
148a
Deduction ($/asset)
Deduct
138
Stabilization network
14
Route orders
140
Stabilization fund unit (S-token)
148b
Execute order
142
146
144
Store transaction details
Update market price

Figure 3

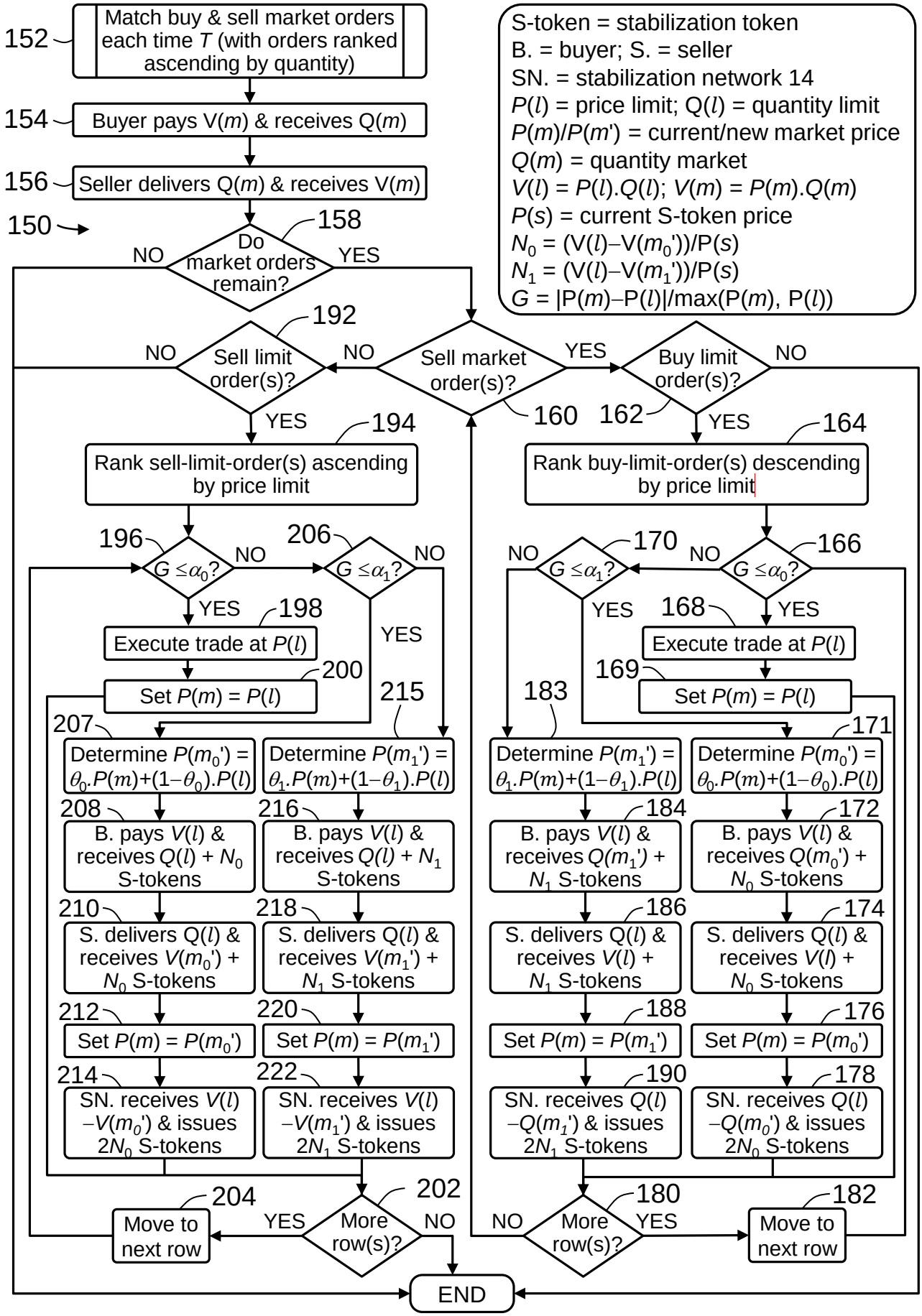

S-token = stabilization token
B. = buyer; S. = seller
SN. = stabilization network 14
P(l) = price limit; Q(l) = quantity limit
P(m)/P(m') = current/new market price
Q(m) = quantity market
V(l) = P(l).Q(l); V(m) = P(m).Q(m)
P(s) = current S-token price
N0 = (V(l)–V(m0'))/P(s)
N1 = (V(l)–V(m1'))/P(s)
G = |P(m)–P(l)|/max(P(m), P(l))
150
152
Match buy & sell market orders each time T (with orders ranked ascending by quantity)
154
Buyer pays V(m) & receives Q(m)
156
Seller delivers Q(m) & receives V(m)
158
Do market orders remain?
NO
YES
160
Sell market order(s)?
162
Buy limit order(s)?
192
Sell limit order(s)?
194
Rank sell-limit-order(s) ascending by price limit
164
Rank buy-limit-order(s) descending by price limit
196
G ≤α0?
206
G ≤α1?
170
G ≤α1?
166
G ≤α0?
198
Execute trade at P(l)
200
Set P(m) = P(l)
168
Execute trade at P(l)
169
Set P(m) = P(l)
215
183
207
Determine P(m0') = θ0.P(m)+(1–θ0).P(l)
Determine P(m1') = θ1.P(m)+(1–θ1).P(l)
Determine P(m1') = θ1.P(m)+(1–θ1).P(l)
171
Determine P(m0') = θ0.P(m)+(1–θ0).P(l)
208
B. pays V(l) & receives Q(l) + N0 S-tokens
216
B. pays V(l) & receives Q(l) + N1 S-tokens
184
B. pays V(l) & receives Q(m1') + N1 S-tokens
172
B. pays V(l) & receives Q(m0') + N0 S-tokens
210
S. delivers Q(l) & receives V(m0') + N0 S-tokens
218
S. delivers Q(l) & receives V(m1') + N1 S-tokens
186
S. delivers Q(l) & receives V(l) + N1 S-tokens
174
S. delivers Q(l) & receives V(l) + N0 S-tokens
212
Set P(m) = P(m0')
220
Set P(m) = P(m1')
188
Set P(m) = P(m1')
176
Set P(m) = P(m0')
214
SN. receives V(l) –V(m0') & issues 2N0 S-tokens
222
SN. receives V(l) –V(m1') & issues 2N1 S-tokens
190
SN. receives Q(l) –Q(m1') & issues 2N1 S-tokens
178
SN. receives Q(l) –Q(m0') & issues 2N0 S-tokens
204
Move to next row
202
More row(s)?
180
More row(s)?
182
Move to next row
END

Figure 4

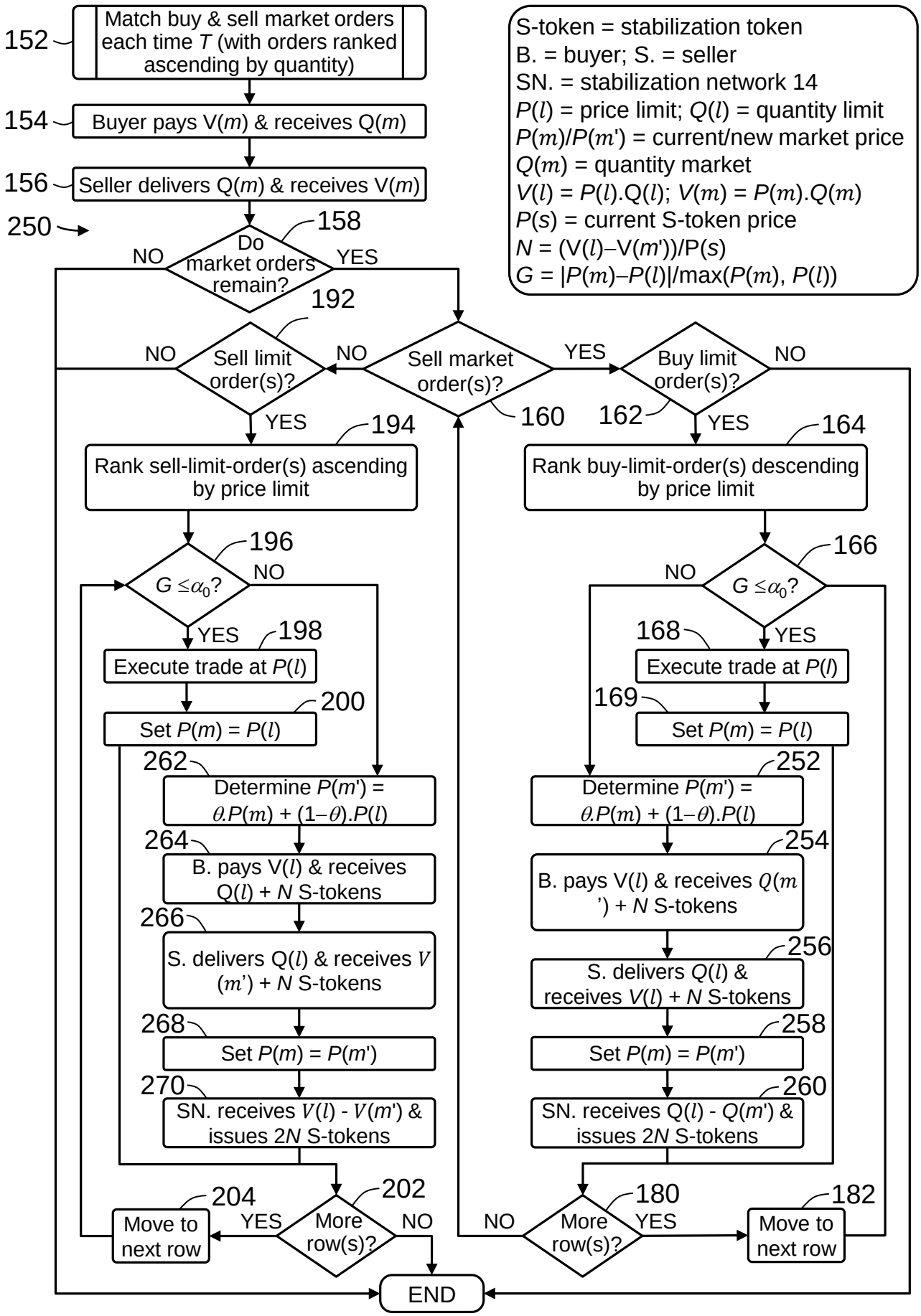

S-token = stabilization token
B. = buyer; S. = seller
SN. = stabilization network 14
P(l) = price limit; Q(l) = quantity limit
P(m)/P(m') = current/new market price
Q(m) = quantity market
V(l) = P(l).Q(l); V(m) = P(m).Q(m)
P(s) = current S-token price
N = (V(l)–V(m'))/P(s)
G = |P(m)–P(l)|/max(P(m), P(l))
152
Match buy & sell market orders each time T (with orders ranked ascending by quantity)
154
Buyer pays V(m) & receives Q(m)
156
Seller delivers Q(m) & receives V(m)
250
158
Do market orders remain?
NO
YES
192
Sell limit order(s)?
NO
Sell market order(s)?
NO
YES
Buy limit order(s)?
NO
YES
160
162
YES
194
Rank sell-limit-order(s) ascending by price limit
164
Rank buy-limit-order(s) descending by price limit
196
G ≤α0?
NO
YES
198
Execute trade at P(l)
200
Set P(m) = P(l)
262
Determine P(m') = θ.P(m) + (1–θ).P(l)
264
B. pays V(l) & receives Q(l) + N S-tokens
266
S. delivers Q(l) & receives V(m') + N S-tokens
268
Set P(m) = P(m')
270
SN. receives V(l) - V(m') & issues 2N S-tokens
204
Move to next row
YES
202
More row(s)?
NO
166
G ≤α0?
NO
YES
168
Execute trade at P(l)
169
Set P(m) = P(l)
252
Determine P(m') = θ.P(m) + (1–θ).P(l)
254
B. pays V(l) & receives Q(m') + N S-tokens
256
S. delivers Q(l) & receives V(l) + N S-tokens
258
Set P(m) = P(m')
260
SN. receives Q(l) - Q(m') & issues 2N S-tokens
180
More row(s)?
NO
YES
182
Move to next row
END

Figure 5