variable data : Type

variables

L0 L1 L2 L3 L4 L5 L6 L7 L8

L9 L10 L11 L12 L13 L14 L15 L16

R0 R1 R2 R3 R4 R5 R6 R7 R8

R9 R10 R11 R12 R13 R14 R15 R16

L0\_D L1\_D L2\_D L3\_D L4\_D L5\_D L6\_D L7\_D L8\_D

L9\_D L10\_D L11\_D L12\_D L13\_D L14\_D L15\_D L16\_D

R0\_D R1\_D R2\_D R3\_D R4\_D R5\_D R6\_D R7\_D R8\_D

R9\_D R10\_D R11\_D R12\_D R13\_D R14\_D R15\_D R16\_D

P1 P2 P3 P4 P5 P6 P7 P8 P9

P10 P11 P12 P13 P14 P15 P16 P17 P18 : data

variable xor (d1 d2 : data) : data

variable F (d : data) : data

definition StepLeft (l r p : data) : data :=

xor r (F (xor l p))

definition StepRight (l r p : data) : data :=

xor l p

definition StepLeftFinal (l r p : data) : data :=

xor r (F (xor l p))

definition StepRightFinal (l r p : data) : data :=

xor l p

premise xor\_comm (x1 x2 : data) :

(xor x1 x2) = (xor x2 x1)

premise xor\_assoc (x1 x2 x3 : data) :

(xor (xor x1 x2) x3) = (xor x1 (xor x2 x3))

premise xor\_cancel : ∀ x1 x2 : data, xor x1 (xor x2 x2) = x1

premise xor\_swap : ∀ x1 x2 x3 : data, x1 = xor x2 x3 → x2 = xor x1 x3

premise L1\_Def : L1 = xor R0 (F (xor L0 P1))

premise R1\_Def : R1 = xor L0 P1

premise L2\_Def : L2 = xor R1 (F (xor L1 P2))

premise R2\_Def : R2 = xor L1 P2

premise L3\_Def : L3 = xor R2 (F (xor L2 P3))

premise R3\_Def : R3 = xor L2 P3

premise L4\_Def : L4 = xor R3 (F (xor L3 P4))

premise R4\_Def : R4 = xor L3 P4

premise L5\_Def : L5 = xor R4 (F (xor L4 P5))

premise R5\_Def : R5 = xor L4 P5

premise L6\_Def : L6 = xor R5 (F (xor L5 P6))

premise R6\_Def : R6 = xor L5 P6

premise L7\_Def : L7 = xor R6 (F (xor L6 P7))

premise R7\_Def : R7 = xor L6 P7

premise L8\_Def : L8 = xor R7 (F (xor L7 P8))

premise R8\_Def : R8 = xor L7 P8

premise L9\_Def : L9 = xor R8 (F (xor L8 P9))

premise R9\_Def : R9 = xor L8 P9

premise L10\_Def : L10 = xor R9 (F (xor L9 P10))

premise R10\_Def : R10 = xor L9 P10

premise L11\_Def : L11 = xor R10 (F (xor L10 P11))

premise R11\_Def : R11 = xor L10 P11

premise L12\_Def : L12 = xor R11 (F (xor L11 P12))

premise R12\_Def : R12 = xor L11 P12

premise L13\_Def : L13 = xor R12 (F (xor L12 P13))

premise R13\_Def : R13 = xor L12 P13

premise L14\_Def : L14 = xor R13 (F (xor L13 P14))

premise R14\_Def : R14 = xor L13 P14

premise L15\_Def : L15 = xor R14 (F (xor L14 P15))

premise R15\_Def : R15 = xor L14 P15

premise L16\_Def : L16 = xor P18 (xor L15 P16)

premise R16\_Def : R16 = xor P17 (xor R15 (F (xor L15 P16)))

premise L0\_D\_Def : L0\_D = L16

premise R0\_D\_Def : R0\_D = R16

premise L1\_D\_Def : L1\_D = xor R0\_D (F (xor L0\_D P18))

premise R1\_D\_Def : R1\_D = xor L0\_D P18

premise L2\_D\_Def : L2\_D = xor R1\_D (F (xor L1\_D P17))

premise R2\_D\_Def : R2\_D = xor L1\_D P17

premise L3\_D\_Def : L3\_D = xor R2\_D (F (xor L2\_D P16))

premise R3\_D\_Def : R3\_D = xor L2\_D P16

premise L4\_D\_Def : L4\_D = xor R3\_D (F (xor L3\_D P15))

premise R4\_D\_Def : R4\_D = xor L3\_D P15

premise L5\_D\_Def : L5\_D = xor R4\_D (F (xor L4\_D P14))

premise R5\_D\_Def : R5\_D = xor L4\_D P14

premise L6\_D\_Def : L6\_D = xor R5\_D (F (xor L5\_D P13))

premise R6\_D\_Def : R6\_D = xor L5\_D P13

premise L7\_D\_Def : L7\_D = xor R6\_D (F (xor L6\_D P12))

premise R7\_D\_Def : R7\_D = xor L6\_D P12

premise L8\_D\_Def : L8\_D = xor R7\_D (F (xor L7\_D P11))

premise R8\_D\_Def : R8\_D = xor L7\_D P11

premise L9\_D\_Def : L9\_D = xor R8\_D (F (xor L8\_D P10))

premise R9\_D\_Def : R9\_D = xor L8\_D P10

premise L10\_D\_Def : L10\_D = xor R9\_D (F (xor L9\_D P9))

premise R10\_D\_Def : R10\_D = xor L9\_D P9

premise L11\_D\_Def : L11\_D = xor R10\_D (F (xor L10\_D P8))

premise R11\_D\_Def : R11\_D = xor L10\_D P8

premise L12\_D\_Def : L12\_D = xor R11\_D (F (xor L11\_D P7))

premise R12\_D\_Def : R12\_D = xor L11\_D P7

premise L13\_D\_Def : L13\_D = xor R12\_D (F (xor L12\_D P6))

premise R13\_D\_Def : R13\_D = xor L12\_D P6

premise L14\_D\_Def : L14\_D = xor R13\_D (F (xor L13\_D P5))

premise R14\_D\_Def : R14\_D = xor L13\_D P5

premise L15\_D\_Def : L15\_D = xor R14\_D (F (xor L14\_D P4))

premise R15\_D\_Def : R15\_D = xor L14\_D P4

premise L16\_D\_Def : L16\_D = xor P1 (xor L15\_D P3)

premise R16\_D\_Def : R16\_D = xor P2 (xor R15\_D (F (xor L15\_D P3)))

theorem ProofL1 : L1\_D = xor P17 R15, :=

have H1 : L1\_D = xor R0\_D (F (xor L0\_D P18)), from L1\_D\_Def,

have H2 : L1\_D = xor R16 (F (xor L0\_D P18)),

from eq.subst R0\_D\_Def H1,

have H3 : L1\_D = xor R16 (F (xor L16 P18)),

from eq.subst L0\_D\_Def H2,

have H4 : L1\_D = xor R16 (F (xor (xor P18 (xor L15 P16)) P18)),

from eq.subst L16\_Def H3,

have H5 : L1\_D = xor R16 (F (xor (xor (xor L15 P16) P18) P18)),

from eq.subst (xor\_comm P18 (xor L15 P16)) H4,

have H6 : L1\_D = xor R16 (F (xor (xor L15 P16) (xor P18 P18))),

from eq.subst (xor\_assoc (xor L15 P16) P18 P18) H5,

have H7 : L1\_D = xor R16 (F (xor L15 P16)),

from eq.subst (xor\_cancel (xor L15 P16) P18) H6,

have H8 : L1\_D = xor (xor P17 (xor R15 (F (xor L15 P16))))

(F (xor L15 P16)),

from eq.subst R16\_Def H7,

have H9 : L1\_D = xor P17 (xor (xor R15 (F (xor L15 P16)))

(F (xor L15 P16)) ),

from eq.subst (xor\_assoc P17

(xor R15 (F (xor L15 P16)))

((F (xor L15 P16)))

) H8,

have H10 : L1\_D = xor P17 (xor R15 (xor (F (xor L15 P16))

(F (xor L15 P16)))),

from eq.subst (xor\_assoc R15

(F (xor L15 P16))

(F (xor L15 P16))

) H9,

show L1\_D = xor P17 R15,

from eq.subst (xor\_cancel R15 (F (xor L15 P16))) H10

theorem ProofR1 : R1\_D = xor L15 P16 :=

have H1 : R1\_D = xor L0\_D P18, from R1\_D\_Def,

have H2 : R1\_D = xor L16 P18, from eq.subst L0\_D\_Def H1,

have H3 : R1\_D = xor (xor P18 (xor L15 P16)) P18,

from eq.subst L16\_Def H2,

have H4 : R1\_D = xor (xor (xor L15 P16) P18) P18,

from eq.subst (xor\_comm P18 (xor L15 P16)) H3,

have H5 : R1\_D = xor (xor L15 P16) (xor P18 P18),

from eq.subst (xor\_assoc (xor L15 P16) P18 P18) H4,

have H6 : R1\_D = xor (xor L15 P16) (xor P18 P18),

from eq.subst (xor\_assoc (xor L15 P16) P18 P18) H5,

show R1\_D = xor L15 P16,

from eq.subst (xor\_cancel (xor L15 P16) P18) H6

theorem ProofStep2

(S1\_L : L1\_D = xor P17 R15) (S1\_R : R1\_D = xor L15 P16) :

(L2\_D = xor (xor L15 P16) (F R15)) ∧ (R2\_D = R15) :=

have H1 : R2\_D = xor L1\_D P17, from R2\_D\_Def,

have H2 : R2\_D = xor (xor P17 R15) P17, from eq.subst S1\_L H1,

have H3 : R2\_D = xor (xor R15 P17) P17,

from eq.subst (xor\_comm P17 R15) H2,

have H4 : R2\_D = xor R15 (xor P17 P17),

from eq.subst (xor\_assoc R15 P17 P17) H3,

have H5 : R2\_D = R15,

from eq.subst (xor\_cancel R15 P17) H4,

have H6 : L2\_D = xor R1\_D (F (xor L1\_D P17)), from L2\_D\_Def,

have H7 : L2\_D = xor R1\_D (F R2\_D),

from eq.subst (eq.symm H1) H6,

have H8 : L2\_D = xor R1\_D (F R15),

from eq.subst H5 H7,

have H9 : L2\_D = xor (xor L15 P16) (F R15),

from eq.subst S1\_R H8,

show (L2\_D = xor (xor L15 P16) (F R15)) ∧ (R2\_D = R15),

from and.intro H9 H5

theorem ProofStep3

(S2\_L : L2\_D = xor (xor L15 P16) (F R15)) (S2\_R : R2\_D = R15) :

(L3\_D = xor R13 P15) ∧ (R3\_D = R14) :=

have H1 : R3\_D = xor L2\_D P16, from R3\_D\_Def,

have H2 : R3\_D = xor (xor (xor L15 P16) (F R15)) P16,

from eq.subst S2\_L H1,

have H3 : R3\_D = xor (xor (F R15) (xor L15 P16)) P16,

from eq.subst (xor\_comm (xor L15 P16) (F R15)) H2,

have H4 : R3\_D = xor (F R15) (xor (xor L15 P16) P16),

from eq.subst (xor\_assoc (F R15) (xor L15 P16) P16) H3,

have H5 : R3\_D = xor (F R15) (xor L15 (xor P16 P16)),

from eq.subst (xor\_assoc L15 P16 P16) H4,

have H6 : R3\_D = xor (F R15) L15,

from eq.subst (xor\_cancel L15 P16) H5,

have H7 : L15 = xor R14 (F R15),

from eq.subst (eq.symm R15\_Def) L15\_Def,

have H8 : R14 = xor L15 (F R15),

from (xor\_swap L15 R14 (F R15)) H7,

have H9 : R14 = xor (F R15) L15,

from eq.subst (xor\_comm L15 (F R15)) H8,

have H10 : R3\_D = R14,

from eq.trans H6 (eq.symm H9),

have H11 : L3\_D = xor R15 (F (xor L2\_D P16)),

from eq.subst S2\_R L3\_D\_Def,

have H12 : L3\_D = xor R15 (F R3\_D),

from eq.subst (eq.symm H1) H11,

have H13 : L3\_D = xor R15 (F R14),

from eq.subst H10 H12,

have H14 : L3\_D = xor (xor L14 P15) (F R14),

from eq.subst R15\_Def H13,

have H15 : L3\_D = xor (xor L14 P15) (F (xor L13 P14)),

from eq.subst R14\_Def H14,

have H16 : L3\_D = xor (xor P15 L14) (F (xor L13 P14)),

from eq.subst (xor\_comm L14 P15) H15,

have H17 : L3\_D = xor P15 (xor L14 (F (xor L13 P14))),

from eq.subst (xor\_assoc P15 L14 (F (xor L13 P14))) H16,

have H18 : R13 = xor L14 (F (xor L13 P14)),

from (xor\_swap L14 R13 (F (xor L13 P14))) L14\_Def,

have H19 : L3\_D = xor P15 R13,

from eq.subst (eq.symm H18) H17,

have H20 : L3\_D = xor R13 P15,

from eq.subst (xor\_comm P15 R13) H19,

show (L3\_D = xor R13 P15) ∧ (R3\_D = R14),

from and.intro H20 H10

theorem ProofStep4

(S3\_R : L3\_D = xor R13 P15) (S3\_L : R3\_D = R14) :

(L4\_D = xor R12 P14) ∧ (R4\_D = R13) :=

have H1 : R4\_D = xor L3\_D P15,

from R4\_D\_Def,

have H2 : R4\_D = xor (xor R13 P15) P15,

from eq.subst S3\_R H1,

have H3 : R4\_D = xor R13 (xor P15 P15),

from eq.subst (xor\_assoc R13 P15 P15) H2,

have H5 : R4\_D = R13,

from eq.subst (xor\_cancel R13 P15) H3,

have H6 : R13 = xor L3\_D P15,

from eq.trans (eq.symm H5) H1,

have H7 : L4\_D = xor R3\_D (F (xor L3\_D P15)),

from L4\_D\_Def,

have H8 : L4\_D = xor R14 (F (xor L3\_D P15)),

from eq.subst S3\_L H7,

have H9 : L4\_D = xor R14 (F R13),

from eq.subst (eq.symm H6) H8,

have H10 : L4\_D = xor (xor L13 P14) (F R13),

from eq.subst R14\_Def H9,

have H11 : L4\_D = xor (xor P14 L13) (F R13),

from eq.subst (xor\_comm L13 P14) H10,

have H12 : L4\_D = xor P14 (xor L13 (F R13)),

from eq.subst (xor\_assoc P14 L13 (F R13)) H11,

have H13 : L4\_D = xor P14 (xor (xor R12 (F (xor L12 P13))) (F R13)),

from eq.subst L13\_Def H12,

have H14 : L4\_D = xor P14 (xor R12 (xor (F (xor L12 P13)) (F R13))),

from eq.subst (xor\_assoc R12 (F (xor L12 P13)) (F R13)) H13,

have H15 : L4\_D = xor P14 (xor R12 (xor (F (xor L12 P13))

(F (xor L12 P13)))),

from eq.subst R13\_Def H14,

have H16 : L4\_D = xor P14 R12,

from eq.subst (xor\_cancel R12 (F (xor L12 P13))) H15,

have H17 : L4\_D = xor R12 P14,

from eq.subst (xor\_comm P14 R12) H16,

show (L4\_D = xor R12 P14) ∧ (R4\_D = R13),

from and.intro H17 H5

theorem ProofStep5Left

(S3\_R : L15\_D = xor R1 P3) (S3\_L : R15\_D = R2) :

L16\_D = L0 :=

have H1 : L16\_D = xor P1 (xor L15\_D P3),

from L16\_D\_Def,

have H2 : L16\_D = xor P1 (xor (xor R1 P3) P3),

from eq.subst S3\_R H1,

have H3 : L16\_D = xor P1 (xor R1 (xor P3 P3)),

from eq.subst (xor\_assoc R1 P3 P3) H2,

have H4 : L16\_D = xor P1 R1,

from eq.subst (xor\_cancel R1 P3) H3,

have H5 : L16\_D = xor P1 (xor L0 P1),

from eq.subst R1\_Def H4,

have H6 : L16\_D = xor (xor L0 P1) P1,

from eq.subst (xor\_comm P1 (xor L0 P1)) H5,

have H7 : L16\_D = xor L0 (xor P1 P1),

from eq.subst (xor\_assoc L0 P1 P1) H6,

show L16\_D = L0,

from eq.subst (xor\_cancel L0 P1) H7

theorem ProofStep5Right

(S3\_R : L15\_D = xor R1 P3) (S3\_L : R15\_D = R2) :

R16\_D = R0 :=

have H1 : R16\_D = xor P2 (xor R15\_D (F (xor L15\_D P3))),

from R16\_D\_Def,

have H2 : R16\_D = xor P2 (xor R15\_D (F (xor (xor R1 P3) P3))),

from eq.subst S3\_R H1,

have H3 : R16\_D = xor P2 (xor R15\_D (F (xor R1 (xor P3 P3)))),

from eq.subst (xor\_assoc R1 P3 P3) H2,

have H4 : R16\_D = xor P2 (xor R15\_D (F R1)),

from eq.subst (xor\_cancel R1 P3) H3,

have H5 : R16\_D = xor P2 (xor R2 (F R1)),

from eq.subst S3\_L H4,

have H6 : R16\_D = xor P2 (xor R2 (F (xor L0 P1))),

from eq.subst R1\_Def H5,

have H7 : R16\_D = xor P2 (xor (xor L1 P2) (F (xor L0 P1))),

from eq.subst R2\_Def H6,

have H8 : R16\_D = xor P2 (xor (xor P2 L1) (F (xor L0 P1))),

from eq.subst (xor\_comm L1 P2) H7,

have H9 : R16\_D = xor P2 (xor P2 (xor L1 (F (xor L0 P1)))),

from eq.subst (xor\_assoc P2 L1 (F (xor L0 P1))) H8,

have H10 : R16\_D = xor (xor P2 P2) (xor L1 (F (xor L0 P1))),

from eq.subst

(eq.symm (xor\_assoc P2 P2 (xor L1 (F (xor L0 P1))))) H9,

have H11 : R16\_D = xor (xor L1 (F (xor L0 P1))) (xor P2 P2),

from eq.subst

(xor\_comm (xor P2 P2) (xor L1 (F (xor L0 P1)))) H10,

have H12 : R16\_D = (xor L1 (F (xor L0 P1))),

from eq.subst

(xor\_cancel (xor L1 (F (xor L0 P1))) P2) H11,

have H13 : R16\_D = (xor (xor R0 (F (xor L0 P1))) (F (xor L0 P1))),

from eq.subst L1\_Def H12,

have H14 : R16\_D = (xor R0 (xor (F (xor L0 P1)) (F (xor L0 P1)))),

from eq.subst (xor\_assoc R0 (F (xor L0 P1)) (F (xor L0 P1))) H13,

show R16\_D = R0,

from eq.subst (xor\_cancel R0 (F (xor L0 P1))) H14