HEX: F8400281

BINARY: 1111\_1000\_0100\_0000\_0000\_0010\_1000\_0001

ARM: LDUR R1, [R20, #0]

HEX : 8B010022

BINARY: 1000\_1011\_0000\_0001 \_ 0000\_0000\_001 0\_0010

ARM: ADD R2, R1, R1

HEX: D1000333

BINARY: 1101\_0001\_0000\_0000\_0000\_0011\_0011\_0011

ARM: SUBI R19, R25, #0

HEX: B40000E3

BINARY: 1011\_0100\_ 0000\_0000\_0000\_0000\_1110\_0011

ARM: CBZ R3, #7

HEX: 91002294

BINARY: 1001\_0001\_0000\_0000\_0010\_0010\_1001\_0100

ARM: ADDI R20, R20, #8

HEX: F81F4281

BINARY: 1111\_1000\_0001\_1111\_0100\_ 0010\_100 0\_0001

ARM: STUR R1, [R20, #500]

HEX: 17FFFFFA

BINARY: 0001\_0111\_1111\_1111\_1111\_1111\_1111\_1010

ARM: B #67108858

BINARY: 1001\_0111\_1111\_1111\_1111\_1111\_1111\_1111

ARM: BL #66129696

BINARY: 0001\_0110\_0001\_1010\_1111\_1011\_1000\_1111

ARM: B #35322767

BINARY: 1011\_0100\_0000\_0000\_0000\_1000\_0010\_0111

ARM: CBZ R7, 65

BINARY: 1011\_0101\_0010\_1001\_0000\_1101\_1100\_0101

ARM: CBNZ R5,18542

BINARY: 1111\_1000\_0100\_0000\_0000\_0101\_0010\_0101

AMR: LDUR R5,[ R9, #1]

BINARY: 1111\_1000\_0001\_1001\_1110\_0010\_1011\_0001

ARM: STUR R17,[R21, #41]

BINARY: 1000\_1011\_0001\_0101\_0000\_0010\_1000\_1001

ARM: ADD R9,R20,R21

BINARY: 1000\_1010\_0001\_0000\_0000\_0110\_0111\_0001

ARM: AND R17,R19,R16

BINARY: 1010\_1010\_0001\_1111\_0001\_0111\_1100\_1100

ARM: ORR R12, R30, R31

BINARY: 1110\_1010\_0000\_1101\_1010\_1111\_1010\_1110

ARM: EOR R14,R29,R13

BINARY: 1100\_1011\_0001\_1000\_0000\_0001\_0010\_1011

ARM: SUB R11, R9, R24

BINARY: 1001\_0001\_0000\_0000\_0000\_0110\_1101\_0110

ARM: ADDI R22, R22, #1

BINARY: 1001\_0001\_0001\_0101\_0000\_0010\_1000\_1001

ARM: ANDI R9, R20, #1344

BINARY: 1011\_0010\_0000\_1000\_1010\_0110\_1010\_1000

ARM: ORRI R8, R21, #276

BINARY: 1101\_0010\_0000\_0000\_1100\_1010\_1000\_1100

ARM : EORI R12, R20, #50

BINARY: 1101\_0001\_0000\_1000\_0110\_1000\_1010\_0111

ARM: SUBI R7, R5, #538

BINARY: 1101\_0010\_1000\_1001\_1110\_1111\_1001\_1000

ARM: MOVZ X24, #20348

LDUR R1, [R20, #0] Memory Address R20 = decimal 5 = 32’b000000000000101 = 0x0000005, R1 GETS what’s in memory 0x00000005 so needs to have 5

ADD R2, R1, R1 R1 = 5 then R2 = 10

SUBI R19, R25, #0 R25 = 15 then R19 = 15

CBZ R3, #7 R3 = 5 stop from branching

ADDI R20, R20, #8 R20 = 5+8 = 13

STUR R1, [R20, #500] 513 = 0x00000201 so we need a block address 0x0201 and R1 stores the value of R1 =5 into 0x0201

B #67108858