List of changes:

* Changed MOV R0, #X to MOV R0, to #60
* Changed LR to PC
* Changed label euqation51 to equation51
* Changed label return to return1
* Changed label divider2 to divide2
* Changed label recusiveSine to recursiveSine
* Changed all the LSR instruction to ASR (totally 11 instructions)

MOV R0, #60

BL cosine

MOV R0, #60

BL doubleAngleSine

MOV R0, #60

BL doubleAngleCosine

cosine PROC

PUSH {R4-R8, R10, R11, LR}

MOV R4, R0

MUL R5, R4, R0

ASR R6, R5, #8 ; LSR -> ASR

MOV R7, #128

SUB R0, R7, R6

POP {R4-R8, R10, R11, PC} ; LR -> PC

ENDP

doubleAngleSine PROC

PUSH {R4-R8, R10, R11, LR}

MOV R4, R0

CMP R4, #-64

BGT checkPos1

BLE equation51 ;euqation51 -> equation

checkPos1 CMP R4, #64

BLT equation31

BGE equation51

equation31 B return1 ;return -> return1

equation51 ASR R4, #1 ; LSR -> ASR

MOV R0, R4

BL cosine

MOV R5, R0

MOV R0, R4

BL doubleAngleSine

MUL R6, R0, R5

ASR R0, R6, #6 ;LSR -> ASR

return1 POP {R4-R8, R10, R11, PC}

ENDP

doubleAngleCosine PROC

PUSH {R4-R8, R10, R11, LR}

MOV R4, R0

CMP R4, #-128

BGT checkPos2

BLE equation72

checkPos2 CMP R4, #128

BLT divide2

BGE equation72

equation72 ASR R0, #1 ;LSR -> ASR

BL doubleAngleSine

MOV R5, R0

MUL R6, R5, R0

ASR R6, #6 ; LSR -> ASR

MOV R7, #128

SUB R0, R7, R6

B return2

divide2 ASR R0, #1 ;divider2 -> divide2 LSR -> ASR

BL cosine

return2 POP {R4-R8, R10, R11, PC}

ENDP

recursiveCosine PROC

PUSH {R4-R8, R10, R11, LR}

MOV R4, R0

CMP R4, #-128

BGT reCheckPos2

BLE reEquation72

reCheckPos2

CMP R4, #128

BLT reDivid2

BGE reEquation72

reEquation72 ASR R0, #1 ;LSR -> ASR

BL doubleAngleSine

MOV R5, R0

MUL R6, R5, R0

ASR R6, #6 ; LSR -> ASR

MOV R7, #128

SUB R0, R7, R6

B reReturn2

reDivid2 ASR R0, #1 ; LSR -> ASR

BL recursiveCosine

reReturn2

POP {R4-R8, R10, R11, PC}

ENDP

recursiveSine PROC ;recusiveSine -> recursiveSine

PUSH {R4-R8, R10, R11, LR}

MOV R4, R0

CMP R4, #-64

BGT reCheckPos1

BLE reEquation51

reCheckPos1 CMP R4, #64

BLT reEquation31

BGE reEquation51

reEquation31 B reReturn1

reEquation51 ASR R4, #1 ; LSR - > ASR

MOV R0, R4

BL recursiveCosine

MOV R5, R0

MOV R0, R4

BL recursiveSine

MUL R6, R0, R5

ASR R0, R6, #6 ; LSR -> ASR

reReturn1

POP {R4-R8, R10, R11, PC}

ENDP