Lab Exercise #11 -- ARM Datapath and Control

Assume that an implementation of the ARM microprocessor only recognizes a subset of the full instruction set. It recognizes the following:

- Data processing instructions (all sixteen)
- Data movement instructions (LDR and STR)
- Branch instructions (B and BL)

The contents of the control and general-purpose registers are shown below (in hexadecimal) after an instruction has been fetched, but before any other actions have occurred.

CPSR: A0000000

R[00]: 00000000  R[04]: 04040404  R[08]: 08080808  R[0C]: 0C0C0C0C
R[01]: 01010101  R[05]: 05050505  R[09]: 09090909  R[0D]: 7FFFFF00
R[02]: 02020202  R[06]: 06060606  R[0A]: 0A0A0A0A  R[0E]: 00108888
R[03]: 03030303  R[07]: 07070707  R[0B]: 0B0B0B0B  R[0F]: 00012000

For each of the following independent cases, assume that the IR (instruction register) contains the value shown (in hexadecimal). Give the hexadecimal value of the requested signals. If the value of a signal cannot be determined from the information given, write "unknown".

a) IR: E00B0005  (and r0, r11, r5)

   ALU input #1: __0B0B0B0B__  (contents of R[B])
   ALU input #2: __05050505__  (contents of R[5])
   ALU output:   __01010101__  (result of AND)
   updated R[rd]: __01010101__  (new contents of R[0] -- condition is "always")
   updated PC:    __00012004__  (PC + 4)

b) IR: 638610CC  (orrvs r1, r6, #0xcc)

   ALU input #1: __06060606__  (contents of R[6])
   ALU input #2: __000000CC__  (immediate value -- Bits 7:0 of the IR)
   ALU output:   __060606CE__  (result of OR)
   updated R[rd]: __01010101__  (R[1] not changed -- condition is false)
   updated PC:    __00012004__  (PC + 4)

c) IR: 40292005  (eormi r2, r9, r5)

   ALU input #1: __09090909__  (contents of R[9])
   ALU input #2: __05050505__  (contents of R[5])
   ALU output:   __0C0C0C0C__  (result of XOR)
   updated R[rd]: __0C0C0C0C__  (new contents of R[2] -- condition is true)
   updated PC:    __00012004__  (PC + 4)
d) IR: E5983004 (ldr r3, [r8, #4])

   ALU input #1: __08080808__ (contents of R[8])
   ALU input #2: __00000004__ (immediate value -- Bits 11:0 of the IR)
   ALU output: __0808080C__ (result of addition)
   updated R[rd]: __unknown__ (contents of RAM[0808080C] is not known)
   updated PC: __00012004__ (PC + 4)

e) IR: E5864002 (str r4, [r6, #2])

   ALU input #1: __06060606__ (contents of R[6])
   ALU input #2: __00000002__ (immediate value -- Bits 11:0 of the IR)
   ALU output: __06060608__ (result of addition)
   updated R[rd]: __no change__ (a Store instruction does not update a register)
   updated PC: __00012004__ (PC + 4)

f) IR: 0A000005 (beq endloop)

   ALU input #1: __unknown__ (a Branch instruction does not use the ALU)
   ALU input #2: __unknown__ (a Branch instruction does not use the ALU)
   ALU output: __unknown__ (a Branch instruction does not use the ALU)
   updated R[rd]: __no change__ (a Branch instruction does not update a register)
   updated PC: __00012004__ (PC + 4, since condition is false)

g) IR: EAFFFFFF6 (b loop)

   ALU input #1: __unknown__ (a Branch instruction does not use the ALU)
   ALU input #2: __unknown__ (a Branch instruction does not use the ALU)
   ALU output: __unknown__ (a Branch instruction does not use the ALU)
   updated R[rd]: __no change__ (a Branch instruction does not update a register)
   updated PC: __00012004__ (PC + 4, since condition is true)

h) IR: EB000009 (bl check)

   ALU input #1: __unknown__ (a Branch instruction does not use the ALU)
   ALU input #2: __unknown__ (a Branch instruction does not use the ALU)
   ALU output: __unknown__ (a Branch instruction does not use the ALU)
   updated R[rd]: __00012000__ (BL stores PC in LR)
   updated PC: __00012024__ (PC + sign_extend(simm24|00), condition is "always")