* 1. Type of instruction: According to the figure A3-1 in ARM manual, ‘Branch and branch with link’ is only one instruction set encoding whose values at [25:27] bit is 101. So, I can figure out this instruction is branch instruction.
  2. Operation – Condition Field: According to the A4.1.5(Page A4-10), there is the detail of the branch instruction. ‘Operation’ part of the instruction said that I should check the condition is passed first. The condition field of this instruction is 1110 and it means the instruction can operate unconditionally.
  3. Operation – L: According to page A4-10, branch instruction branches without storing a return address when L is omitted. In the case of this instruction, it doesn’t need to store any return address because the L bit is 0.
  4. Operation – Target Address: According to page A4-10 in ARM manual, the target address is calculated like below.
     1. First, the result of sign-extending the 24-bit signed immediate to 30 bits is 00 0000 0000 0000 0000 0000 1010 0111. (Because the signed immediate is 0000 0000 0000 0000 1010 0111 here.)
     2. Then, get 0000 0000 0000 0000 0000 0010 1001 1100 by shifting the result left two bits.
     3. Because the address of this instruction is 2, the content of PC will be 2\*4 + 8 bytes. So, the target address will be (2\*4+8) + 668 = 684(bytes). It means after the operation of this instruction, PC will be move to 684/4 = 171 = 0xAB.
     4. Therefore, I can write the assembly code of this instruction like ‘B #0xAB;’ because the syntax of branch instruction is ‘B{L}{cond} <target\_address>’.
  5. Type of instruction: According to the figure A3-1 in ARM manual, ‘Branch and branch with link’ is only one instruction set encoding whose values at [25:27] bit is 101. So, I can figure out this instruction is branch instruction.
  6. Operation – Condition Field: According to the A4.1.5(Page A4-10), there is the detail of the branch instruction. ‘Operation’ part of the instruction said that I should check the condition is passed first. The condition field of this instruction is 1110 and it means the instruction can operate unconditionally.
  7. Operation – L: According to page A4-10, branch instruction branches without storing a return address when L is omitted. In the case of this instruction, it doesn’t need to store any return address because the L bit is 0.
  8. Operation – Target Address: According to page A4-10 in ARM manual, the target address is calculated like below.
     1. First, the result of sign-extending the 24-bit signed immediate to 30 bits is 00 0000 0000 0000 0000 0000 1010 0111. (Because the signed immediate is 0000 0000 0000 0000 1010 0111 here.)
     2. Then, get 0000 0000 0000 0000 0000 0010 1001 1100 by shifting the result left two bits.
     3. Because the address of this instruction is 2, the content of PC will be 2\*4 + 8 bytes. So, the target address will be (2\*4+8) + 668 = 684(bytes). It means after the operation of this instruction, PC will be move to 684/4 = 171 = 0xAB.
     4. Therefore, I can write the assembly code of this instruction like ‘B #0xAB;’ because the syntax of branch instruction is ‘B{L}{cond} <target\_address>’.