**Computer Architecture Assignment #1**

Your Student Number: 2023320046

Name: 정승우

<<Notice>>

* Please edit the title of this document correctly.
  + (ex. CA1\_2023012345\_Tom-Cruise OR CA1\_2023012345\_홍길동)
* Please write your information(Student number and name) correctly.
* You can write your answers in English or Korean.
* Don’t change the layout(colored in Black) of this document. (Please edit just blue-colored part.)
* Before submitting, don’t forget to convert the file to a PDF format.

**Address 000 | Instruction EA000006 (Example)**

1. Change to binary format: 1110 1010 0000 0000 0000 0000 0000 0110
2. Write assembly code: B #8;
3. Describe why you wrote the assembly code like above:
   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 0000 0000 0000 0110. (Because the signed immediate is 0000 0000 0000 0000 0000 0110 here.)
      2. Then, get 0000 0000 0000 0000 0000 0000 0001 1000 by shifting the result left two bits.
      3. Because the address of this instruction is 0, the content of PC will be 0 + 8 bytes. So, the target address will be (0+8) + 24 = 32(bytes). It means after the operation of this instruction, PC will be move to 32/4 = 8.
      4. Therefore, I can write the assembly code of this instruction like ‘B #8;’ because the syntax of branch instruction is ‘B{L}{cond} <target\_address>’.
4. What is the meaning of the instruction? : The instruction means ‘branch to address 8’.

Address 001 | Instruction EAFFFFFE

1. Change to binary format: 1110 1010 1111 1111 1111 1111 1111 1110
2. Write assembly code: B #0x1;
3. Describe why you wrote the assembly code like above:
   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 11 1111 1111 1111 1111 1111 1111 1110. (Because the signed immediate is 1111 1111 1111 1111 1111 1110 here.)
      2. Then, get 1111 1111 1111 1111 1111 1111 1111 1000 by shifting the result left two bits.
      3. Because the address of this instruction is 1, the content of PC will be 1\*4 + 8 bytes. So, the target address will be (1\*4+8) + -8 = 4(bytes). It means after the operation of this instruction, PC will be move to 4/4 = 1.
      4. Therefore, I can write the assembly code of this instruction like ‘B #1;’ because the syntax of branch instruction is ‘B{L}{cond} <target\_address>’.
4. What is the meaning of the instruction? : The instruction means ‘branch to address 1’.

Address 002 | Instruction EA0000A7

1. Change to binary format: 1110 1010 0000 0000 0000 0000 1010 0111
2. Write assembly code: B #0xAB;
3. Describe why you wrote the assembly code like above:
   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>’.
4. What is the meaning of the instruction? : The instruction means ‘branch to address 0xAB’.

Address 003~005 | Instruction EAFFFFFE

1. Change to binary format: 1110 1010 1111 1111 1111 1111 1111 1110
2. Write assembly code: B #0x3; / B #0x4; / B #0x5;
3. Describe why you wrote the assembly code like above:
   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 11 1111 1111 1111 1111 1111 1111 1110. (Because the signed immediate is 1111 1111 1111 1111 1111 1110 here.)
      2. Then, get 1111 1111 1111 1111 1111 1111 1111 1000 by shifting the result left two bits.
      3. Because the address of this instruction is 1, the content of PC will be PC\*4 + 8 bytes. So, the target address will be (PC\*4+8) + -8 = 4\*PC(bytes). It means after the operation of this instruction, PC will be move to 4\*PC/4 = PC.
4. What is the meaning of the instruction? : The instruction means ‘branch to current PC’.

Address 006 | Instruction EA0000A4

1. Change to binary format: 1110 1010 0000 0000 0000 0000 1010 0100
2. Write assembly code: B #0xAC;
3. Describe why you wrote the assembly code like above:
   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 0100. (Because the signed immediate is 0000 0000 0000 0000 1010 0100 here.)
      2. Then, get 0000 0000 0000 0000 0000 0010 1001 0000 by shifting the result left two bits.
      3. Because the address of this instruction is 6, the content of PC will be 6\*4 + 8 bytes. So, the target address will be 6 \* 4 + 8 + 4\*(4+ 32 + 128) = 688(bytes). It means after the operation of this instruction, PC will be move to 688/4 = 172 = 0xAC.
4. What is the meaning of the instruction? : The instruction means ‘branch to address 0xAC’.

Address 007 | Instruction EAFFFFFE

1. Change to binary format: 1110 1010 1111 1111 1111 1111 1111 1110
2. Write assembly code: B #0x7;
3. Describe why you wrote the assembly code like above:
   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 11 1111 1111 1111 1111 1111 1111 1110. (Because the signed immediate is 1111 1111 1111 1111 1111 1110 here.)
      2. Then, get 1111 1111 1111 1111 1111 1111 1111 1000 by shifting the result left two bits.
      3. Because the address of this instruction is 1, the content of PC will be PC\*4 + 8 bytes. So, the target address will be (PC\*4+8) + -8 = 4\*PC(bytes). It means after the operation of this instruction, PC will be move to 4\*PC/4 = PC.
4. What is the meaning of the instruction? : The instruction means ‘branch to current PC’.

Address 008 | Instruction E59F2EC8

1. Change to binary format: 1110 0101 1001 1111 0010 1110 1100 1000
2. Write assembly code: LDR R2 [R15, #0xEC8];
3. Describe why you wrote the assembly code like above:
   1. Type of instruction: According to the page A5-20 in ARM manual, ‘Load and Store Word or Unsigned Byte - Immediate offset’ is only one instruction set encoding whose values at [27:24] bit is 0101 and 21 bit is 0. So, I can figure out this instruction is Load and Store Word or Unsigned Byte - Immediate offset.
   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 – P: According to page A5-19, indicates whether the addressing mode is post-indexed addressing, or pre-indexed addressing. The memory address is generated by applying the offset to the base register value since (P == 1)
   4. Operation – U: According to page A5-20, indicates whether the offset is added to the base (U == 1) or is subtracted from the base (U == 0). In this instruction, the offset is added to the base since (U == 1)
   5. Operation – B: According to page A3-22, distinguishes between an unsigned byte (B==1) and a word (B==0) access. This instruction is an word access since B == 1
   6. Operation – W: According to page A5-19, if W == 0, the base register is not updated (offset addressing). if W == 1, the calculated memory address is written back to the base register (pre-indexed addressing) since (P == 1). Thus, the base register is not updated since (W == 1)
   7. Operation – L: According to page A3-22, distinguishes between a Load (L==1) and a Store instruction (L==0). This instruction is Load instruction since L == 1.
   8. Operation – Rn: According to page A3-22, value of Rn is [19:16] bit, which is 1111 = F (PC Register).
   9. Operation – Rd: According to page A3-22, value of Rd is [15:12] bit, which is 0010 = 2.
   10. Operation – Offset: According to page A3-21 in ARM manual, the offset is an unsigned number that can be added to or subtracted from the base register, and its value is [11:0] = 1110 1100 1000. Thus, the Target Address will be calculated with R[Rn] + unsigned\_extension(imm12) = R[15] + 0xEC8.
4. What is the meaning of the instruction? : ‘Load data from memory address (R[15] + 0xEC8.) and write to R[2]’

Address 009 | Instruction E3A00040

1. Change to binary format: 1110 0011 1010 0000 0000 0000 0100 0000
2. Write assembly code: MOV R0, #64;
3. Describe why you wrote the assembly code like above:
   1. Type of instruction: According to the page A3-2, the instruction is either “Data processing immediate”, or “Move immediate to status register”. From page A4-76, this instruction does not match with MSR(Move to Status Register from ARM Register), this instruction is “Data processing immediate”.
   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 – opcode : According to the page A3-7, the [24:21] bit is instruction opcode, which is 1101, is MOV instruction. Thus, this instruction is MOV instruction.
   4. Operation – S : According to the page A3-8, [20] bit is S bit, which signifies that the instruction updates the condition codes. This instruction does not update the condition codes since [20] bit is 0, it does not update the condition code.
   5. Operation – Rn: According to page A3-8, value of Rn is [19:16] bit, which is 0000. Also, MOV bit does not take first operand, Rn, thus Rn is negligible.
   6. Operation – Rd: According to page A3-8, value of Rd is [15:12] bit, which is 0000 = 0.
   7. Operation – shifter\_operand: According to page A5-3, since [27:25] bit is 001, the shifter\_operand is rotate\_imm and immed\_8. According to page A5-6, the immediate value is calculated like below.
      1. First, immediate value is immed\_8 Rotate right (rotate \* 2)
      2. Rotate\_imm value is [11:8] bit, which is 0000.
      3. Immed\_8 value is [7:0] bit, which is 0100 0000 = 64.
      4. Thus, the immediate value is 64 rotate right 0\*2 = 64.
4. What is the meaning of the instruction? : MOV R[0], 64.

Address 00A | Instruction E5820010

1. Change to binary format: 1110 0101 1000 0010 0000 0000 0001 0000
2. Write assembly code: STR R0, [R2, #16];
3. Describe why you wrote the assembly code like above:
   1. Type of instruction: According to the page A5-20 in ARM manual, ‘Load and Store Word or Unsigned Byte - Immediate offset’ is only one instruction set encoding whose values at [27:24] bit is 0101 and 21 bit is 0. So, I can figure out this instruction is Load and Store Word or Unsigned Byte - Immediate offset.
   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 – P: According to page A5-19, indicates whether the addressing mode is post-indexed addressing, or pre-indexed addressing. The memory address is generated by applying the offset to the base register value since (P == 1)
   4. Operation – U: According to page A5-20, indicates whether the offset is added to the base (U == 1) or is subtracted from the base (U == 0). In this instruction, the offset is added to the base since (U == 1)
   5. Operation – B: According to page A3-22, distinguishes between an unsigned byte (B==1) and a word (B==0) access. This instruction is an word access since B == 1
   6. Operation – W: According to page A5-19, if W == 0, the base register is not updated (offset addressing). if W == 1, the calculated memory address is written back to the base register (pre-indexed addressing) since (P == 1). Thus, the base register is not updated since (W == 1)
   7. Operation – L: According to page A3-22, distinguishes between a Load (L==1) and a Store instruction (L==0). This instruction is Store instruction since L == 0.
   8. Operation – Rn: According to page A3-22, value of Rn is [19:16] bit, which is 0010 = 2.
   9. Operation – Rd: According to page A3-22, value of Rd is [15:12] bit, which is 0000 = 0.
   10. Operation – Offset: According to page A3-21 in ARM manual, the offset is an unsigned number that can be added to or subtracted from the base register, and its value is [11:0] =. 0000 0001 0000. Thus, the Target Address will be calculated with R[Rn] + unsigned\_extension(imm12) = R[2] + 0x010.
4. What is the meaning of the instruction? : Store R[0], MEM[R2 + 16]

Address 00B | Instruction E5820014

1. Change to binary format: 1110 0101 1000 0010 0000 0000 0001 0100
2. Write assembly code: STR R0, [R2, #20];
   1. Describe why you wrote the assembly code like above: Type of instruction: According to the page A5-20 in ARM manual, ‘Load and Store Word or Unsigned Byte - Immediate offset’ is only one instruction set encoding whose values at [27:24] bit is 0101 and 21 bit is 0. So, I can figure out this instruction is Load and Store Word or Unsigned Byte - Immediate offset.
   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 – P: According to page A5-19, indicates whether the addressing mode is post-indexed addressing, or pre-indexed addressing. The memory address is generated by applying the offset to the base register value since (P == 1)
   4. Operation – U: According to page A5-20, indicates whether the offset is added to the base (U == 1) or is subtracted from the base (U == 0). In this instruction, the offset is added to the base since (U == 1)
   5. Operation – B: According to page A3-22, distinguishes between an unsigned byte (B==1) and a word (B==0) access. This instruction is an word access since B == 1
   6. Operation – W: According to page A5-19, if W == 0, the base register is not updated (offset addressing). if W == 1, the calculated memory address is written back to the base register (pre-indexed addressing) since (P == 1). Thus, the base register is not updated since (W == 1)
   7. Operation – L: According to page A3-22, distinguishes between a Load (L==1) and a Store instruction (L==0). This instruction is Store instruction since L == 0.
   8. Operation – Rn: According to page A3-22, value of Rn is [19:16] bit, which is 0010 = 2.
   9. Operation – Rd: According to page A3-22, value of Rd is [15:12] bit, which is 0000 = 0.
   10. Operation – Offset: According to page A3-21 in ARM manual, the offset is an unsigned number that can be added to or subtracted from the base register, and its value is [11:0] =. 0000 0001 0100. Thus, the Target Address will be calculated with R[Rn] + unsigned\_extension(imm12) = R[2] + 0x014.
3. What is the meaning of the instruction? : Store R[0], MEM[R2 + 20]

Address 00C | Instruction E5820018

1. Change to binary format: 1110 0101 1000 0010 0000 0000 0001 1000
2. Write assembly code: STR R0, [R2, #24];
3. Describe why you wrote the assembly code like above:
   1. Describe why you wrote the assembly code like above: Type of instruction: According to the page A5-20 in ARM manual, ‘Load and Store Word or Unsigned Byte - Immediate offset’ is only one instruction set encoding whose values at [27:24] bit is 0101 and 21 bit is 0. So, I can figure out this instruction is Load and Store Word or Unsigned Byte - Immediate offset.
   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 – P: According to page A5-19, indicates whether the addressing mode is post-indexed addressing, or pre-indexed addressing. The memory address is generated by applying the offset to the base register value since (P == 1)
   4. Operation – U: According to page A5-20, indicates whether the offset is added to the base (U == 1) or is subtracted from the base (U == 0). In this instruction, the offset is added to the base since (U == 1)
   5. Operation – B: According to page A3-22, distinguishes between an unsigned byte (B==1) and a word (B==0) access. This instruction is an word access since B == 1
   6. Operation – W: According to page A5-19, if W == 0, the base register is not updated (offset addressing). if W == 1, the calculated memory address is written back to the base register (pre-indexed addressing) since (P == 1). Thus, the base register is not updated since (W == 1)
   7. Operation – L: According to page A3-22, distinguishes between a Load (L==1) and a Store instruction (L==0). This instruction is Store instruction since L == 0.
   8. Operation – Rn: According to page A3-22, value of Rn is [19:16] bit, which is 0010 = 2.
   9. Operation – Rd: According to page A3-22, value of Rd is [15:12] bit, which is 0000 = 0.
   10. Operation – Offset: According to page A3-21 in ARM manual, the offset is an unsigned number that can be added to or subtracted from the base register, and its value is [11:0] =. 0000 0001 1000. Thus, the Target Address will be calculated with R[Rn] + unsigned\_extension(imm12) = R[2] + 0x018.
4. What is the meaning of the instruction? : Store R[0], MEM[R2 + 24]

Address 00D | Instruction E582001C

1. Change to binary format: 1110 0101 1000 0010 0000 0000 0001 1100
2. Write assembly code: STR R0, [R2, #28];
3. Describe why you wrote the assembly code like above:
   1. Type of instruction: According to the page A5-20 in ARM manual, ‘Load and Store Word or Unsigned Byte - Immediate offset’ is only one instruction set encoding whose values at [27:24] bit is 0101 and 21 bit is 0. So, I can figure out this instruction is Load and Store Word or Unsigned Byte - Immediate offset.
   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 – P: According to page A5-19, indicates whether the addressing mode is post-indexed addressing, or pre-indexed addressing. The memory address is generated by applying the offset to the base register value since (P == 1)
   4. Operation – U: According to page A5-20, indicates whether the offset is added to the base (U == 1) or is subtracted from the base (U == 0). In this instruction, the offset is added to the base since (U == 1)
   5. Operation – B: According to page A3-22, distinguishes between an unsigned byte (B==1) and a word (B==0) access. This instruction is an word access since B == 1
   6. Operation – W: According to page A5-19, if W == 0, the base register is not updated (offset addressing). if W == 1, the calculated memory address is written back to the base register (pre-indexed addressing) since (P == 1). Thus, the base register is not updated since (W == 1)
   7. Operation – L: According to page A3-22, distinguishes between a Load (L==1) and a Store instruction (L==0). This instruction is Store instruction since L == 0.
   8. Operation – Rn: According to page A3-22, value of Rn is [19:16] bit, which is 0010 = 2.
   9. Operation – Rd: According to page A3-22, value of Rd is [15:12] bit, which is 0000 = 0.
   10. Operation – Offset: According to page A3-21 in ARM manual, the offset is an unsigned number that can be added to or subtracted from the base register, and its value is [11:0] =. 0000 0001 1100. Thus, the Target Address will be calculated with R[Rn] + unsigned\_extension(imm12) = R[2] + 0x01C.
4. What is the meaning of the instruction? : Store R[R0], MEM[R2 + 28]

Address 00E | Instruction E5820020

1. Change to binary format: 1110 0101 1000 0010 0000 0000 0010 0000
2. Write assembly code: STR R0, [R2, #32];
3. Describe why you wrote the assembly code like above:
   1. Type of instruction: According to the page A5-20 in ARM manual, ‘Load and Store Word or Unsigned Byte - Immediate offset’ is only one instruction set encoding whose values at [27:24] bit is 0101 and 21 bit is 0. So, I can figure out this instruction is Load and Store Word or Unsigned Byte - Immediate offset.
   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 – P: According to page A5-19, indicates whether the addressing mode is post-indexed addressing, or pre-indexed addressing. The memory address is generated by applying the offset to the base register value since (P == 1)
   4. Operation – U: According to page A5-20, indicates whether the offset is added to the base (U == 1) or is subtracted from the base (U == 0). In this instruction, the offset is added to the base since (U == 1)
   5. Operation – B: According to page A3-22, distinguishes between an unsigned byte (B==1) and a word (B==0) access. This instruction is an word access since B == 1
   6. Operation – W: According to page A5-19, if W == 0, the base register is not updated (offset addressing). if W == 1, the calculated memory address is written back to the base register (pre-indexed addressing) since (P == 1). Thus, the base register is not updated since (W == 1)
   7. Operation – L: According to page A3-22, distinguishes between a Load (L==1) and a Store instruction (L==0). This instruction is Store instruction since L == 0.
   8. Operation – Rn: According to page A3-22, value of Rn is [19:16] bit, which is 0010 = 2.
   9. Operation – Rd: According to page A3-22, value of Rd is [15:12] bit, which is 0000 = 0.
   10. Operation – Offset: According to page A3-21 in ARM manual, the offset is an unsigned number that can be added to or subtracted from the base register, and its value is [11:0] =. 0000 0010 0000. Thus, the Target Address will be calculated with R[Rn] + unsigned\_extension(imm12) = R[2] + 0x020.
4. What is the meaning of the instruction? : Store R[R0], MEM[R2 + 32]

Address 00F | Instruction E5820024

1. Change to binary format: 1110 0101 1000 0010 0000 0000 0010 0100
2. Write assembly code: STR R0, [R2, #36];
3. Describe why you wrote the assembly code like above:
   1. Type of instruction: According to the page A5-20 in ARM manual, ‘Load and Store Word or Unsigned Byte - Immediate offset’ is only one instruction set encoding whose values at [27:24] bit is 0101 and 21 bit is 0. So, I can figure out this instruction is Load and Store Word or Unsigned Byte - Immediate offset.
   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 – P: According to page A5-19, indicates whether the addressing mode is post-indexed addressing, or pre-indexed addressing. The memory address is generated by applying the offset to the base register value since (P == 1)
   4. Operation – U: According to page A5-20, indicates whether the offset is added to the base (U == 1) or is subtracted from the base (U == 0). In this instruction, the offset is added to the base since (U == 1)
   5. Operation – B: According to page A3-22, distinguishes between an unsigned byte (B==1) and a word (B==0) access. This instruction is an word access since B == 1
   6. Operation – W: According to page A5-19, if W == 0, the base register is not updated (offset addressing). if W == 1, the calculated memory address is written back to the base register (pre-indexed addressing) since (P == 1). Thus, the base register is not updated since (W == 1)
   7. Operation – L: According to page A3-22, distinguishes between a Load (L==1) and a Store instruction (L==0). This instruction is Store instruction since L == 0.
   8. Operation – Rn: According to page A3-22, value of Rn is [19:16] bit, which is 0010 = 2.
   9. Operation – Rd: According to page A3-22, value of Rd is [15:12] bit, which is 0000 = 0.
   10. Operation – Offset: According to page A3-21 in ARM manual, the offset is an unsigned number that can be added to or subtracted from the base register, and its value is [11:0] =. 0000 0010 0100. Thus, the Target Address will be calculated with R[Rn] + unsigned\_extension(imm12) = R[2] + 0x024.
4. What is the meaning of the instruction? : Store R[R0], MEM[R2 + 36]

Address 010 | Instruction E3A0003F

1. Change to binary format: 1110 0011 1010 0000 0000 0000 0011 1111
2. Write assembly code: MOV R0, #63;
3. Describe why you wrote the assembly code like above:
   1. Type of instruction: According to the page A3-2, the instruction is either “Data processing immediate”, or “Move immediate to status register”. From page A4-76, this instruction does not match with MSR(Move to Status Register from ARM Register), this instruction is “Data processing immediate”.
   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 – opcode : According to the page A3-7, the [24:21] bit is instruction opcode, which is 1101, is MOV instruction. Thus, this instruction is MOV instruction.
   4. Operation – S : According to the page A3-8, [20] bit is S bit, which signifies that the instruction updates the condition codes. This instruction does not update the condition codes since [20] bit is 0, it does not update the condition code.
   5. Operation – Rn: According to page A3-8, value of Rn is [19:16] bit, which is 0000. Also, MOV bit does not take first operand, Rn, thus Rn is negligible.
   6. Operation – Rd: According to page A3-8, value of Rd is [15:12] bit, which is 0000 = 0.
   7. Operation – shifter\_operand: According to page A5-3, since [27:25] bit is 001, the shifter\_operand is rotate\_imm and immed\_8. According to page A5-6, the immediate value is calculated like below.
      1. First, immediate value is immed\_8 Rotate right (rotate \* 2)
      2. Rotate\_imm value is [11:8] bit, which is 0000.
      3. Immed\_8 value is [7:0] bit, which is 0100 0000 = 64.
      4. Thus, the immediate value is 64 rotate right 0\*2 = 64.
4. What is the meaning of the instruction? : MOV R[0], 64.
5. What is the meaning of the instruction? :

Address 011 | Instruction E5820028

1. Change to binary format: 1110 0101 1000 0010 0000 0000 0010 1000
2. Write assembly code: STR R0, [R2, #40];
3. Describe why you wrote the assembly code like above:
4. What is the meaning of the instruction? :

Address 012 | Instruction E3A00008

1. Change to binary format: 1110 0011 1010 0000 0000 0000 0000 1000
2. Write assembly code: MOV R0, #8;
3. Describe why you wrote the assembly code like above:
4. What is the meaning of the instruction? :

Address 013 | Instruction E582002C

1. Change to binary format: 1110 0101 1000 0010 0000 0000 0010 1100
2. Write assembly code: STR R0, [R2, #44];
3. Describe why you wrote the assembly code like above:
4. What is the meaning of the instruction? :

Address 014 | Instruction E59F3E9C

1. Change to binary format: 1110 0101 1001 1111 0011 1110 1001 1100
2. Write assembly code: LDR R3, [R15, #0xE9C];
3. Describe why you wrote the assembly code like above:
4. What is the meaning of the instruction? :

Address 015 | Instruction E59F1E9C

1. Change to binary format: 1110 0101 1001 1111 0001 1110 1001 1100
2. Write assembly code: LDR R1, [R15, #0xE9C];
3. Describe why you wrote the assembly code like above:
4. What is the meaning of the instruction? :

Address 016 | Instruction E5831000

1. Change to binary format: 1110 0101 1000 0011 0001 0000 0000 0000
2. Write assembly code: STR R1, R3;
3. Describe why you wrote the assembly code like above:
4. What is the meaning of the instruction? :

Address 017 | Instruction E59F9E98

1. Change to binary format: 1110 0101 1001 1111 1001 1110 1001 1000
2. Write assembly code: LDR R9, [R15, #0xE98];
3. Describe why you wrote the assembly code like above:
4. What is the meaning of the instruction? :

Address 018 | Instruction E3A08000

1. Change to binary format: 1110 0011 1010 0000 1000 0000 0000 0000
2. Write assembly code: MOV R8, #0;
3. Describe why you wrote the assembly code like above:
4. What is the meaning of the instruction? :

Address 019 | Instruction E5898000

1. Change to binary format: 1110 0101 1000 1001 1000 0000 0000 0000
2. Write assembly code: STR R8, R9;
3. Describe why you wrote the assembly code like above:
4. What is the meaning of the instruction? :

Address 01A | Instruction E5898004

1. Change to binary format: 1110 0101 1000 1001 1000 0000 0000 0100
2. Write assembly code: STR R8, [R9, #4];
3. Describe why you wrote the assembly code like above:
4. What is the meaning of the instruction? :

Address 01B | Instruction E5898008

1. Change to binary format: 1110 0101 1000 1001 1000 0000 0000 1000
2. Write assembly code: STR R8, [R9, #8];
3. Describe why you wrote the assembly code like above:
4. What is the meaning of the instruction? :

Address 01C | Instruction E589800C

1. Change to binary format: 1110 0101 1000 1001 1000 0000 0000 1100
2. Write assembly code: STR R8, [R9, #12];
3. Describe why you wrote the assembly code like above:
4. What is the meaning of the instruction? :

Address 01D | Instruction E5898010

1. Change to binary format: 1110 0101 1000 1001 1000 0000 0001 0000
2. Write assembly code: STR R8, [R9, #16];
3. Describe why you wrote the assembly code like above:
4. What is the meaning of the instruction? :

Address 01E | Instruction E5898014

1. Change to binary format: 1110 0101 1000 1001 1000 0000 0001 0100
2. Write assembly code: STR R8, [R9, #20];
3. Describe why you wrote the assembly code like above:
4. What is the meaning of the instruction? :

Address 01F | Instruction E5898018

1. Change to binary format: 1110 0101 1000 1001 1000 0000 0001 1000
2. Write assembly code: STR R8, [R9, #24];
3. Describe why you wrote the assembly code like above:
4. What is the meaning of the instruction? :

Address 020 | Instruction E59FDE78

1. Change to binary format: 1110 0101 1001 1111 1101 1110 0111 1000
2. Write assembly code: LDR R13, [R15, #0xE78];
3. Describe why you wrote the assembly code like above:
4. What is the meaning of the instruction? :

Address 021 | Instruction E5931200

1. Change to binary format: 1110 0101 1001 0011 0001 0010 0000 0000
2. Write assembly code: LDR R1, [R3, #0x200];
3. Describe why you wrote the assembly code like above:
4. What is the meaning of the instruction? :

Address 022 | Instruction E3510001

1. Change to binary format: 1110 0011 0101 0001 0000 0000 0000 0001
2. Write assembly code: CMP R1, #1;
3. Describe why you wrote the assembly code like above:
4. What is the meaning of the instruction? :

Address 023 | Instruction 0A000000

1. Change to binary format: 0000 1010 0000 0000 0000 0000 0000 0000
2. Write assembly code: BEQ #0x25;
3. Describe why you wrote the assembly code like above:
4. What is the meaning of the instruction? :

Address 024 | Instruction EAFFFFFB

1. Change to binary format: 1110 1010 1111 1111 1111 1111 1111 1011
2. Write assembly code: B #0x21;
3. Describe why you wrote the assembly code like above:
4. What is the meaning of the instruction? :

Explain the actual execution flow of the instructions(Address 000~024)

Specify where the execution ends (If not, specify the range repeated in detail)