Computer Organization and Architecture
Prof. V. Kamakoti
Department of Computer Science and Engineering
Indian Institute of Technology, Madras

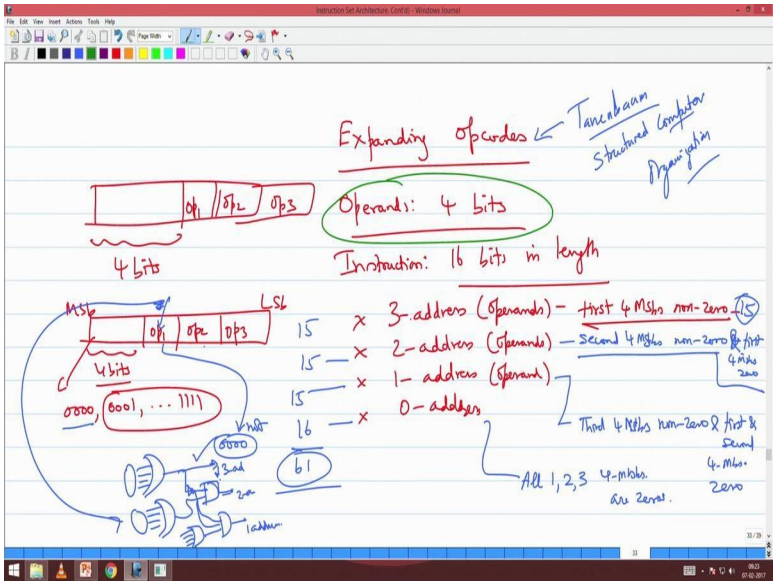
Lecture - 16
Expanding Opcodes

Good morning.

Student: Good morning.

Good morning. So, we start off with one more point to be added in the case of instruction set design this concept is called expanding opcodes.

(Refer Slide Time: 00:33)



Suppose I need; I have operands which can be addressed using 4 bits and I have instructions which can be 16 bits in length. So, how many 3 address 2 address one address

I mean 3 addresses means 3 operands 2 operands one operand and 0 address and 0 operands; how many instructions can you design. So, I have a how many instructions you can have right maximize on this suppose I have 16 bit instruction length and each operand will take 4 bits let us say I have 16 general purpose register each operand will take 4 bits to address it now how many 3 address 2 address one address and 0 address

instructions you can construct is there a systematic way of doing it do you understand the question yes or no what yeah.

Student: Sir, how many case of 3 address will use all of 4; 4 bits like 16 bit 16 bit instructions.

But the instructions are constant length, no; I cannot adjust in that every instruction is 16 bits in length;

Student: So, I am not sure, but I think if it is just one address bits then we can specify more than one instruction in 16 bit.

But I need in this priority I need more 3 address less 2 address one address and so on.

Student: Sir will not depend on size of the opcode like.

So, total instruction size is 16 bits and I need 3; 3 address 2 address one address and 0 address opcode can be whatever right opcode does not need any it can if you have k bits I could have 2 power k opcodes right I need the verity of all these things, but I would like to maximize 3 address 2 address. So, what is systematic way of doing it very quickly and why should we do it yes at the end the decoding should also be very simple as I told you there are 2 translations involved right compiler translates to machine language that is an explicit translation which you see with your eyes the machine language is again interpreted by the hardware. So, that is second interpretations are you getting this. So, you should get me a way; by which I can design. So, that my decoding also becomes simple.

Student: First 4 bits which we can use for opcode generation.

First 4 bit we can use for opcode yeah, yeah that is obvious then.

Student: Then if there is 3 operands; we use the next one or all the remaining we can use 4; 4 each we can use it for.

3 operand, then;

Student: For 2 operand; we will leave 0 0 0 0.

How will the hardware know it is 2 operand or 3 operand?

Student: Might need 2 make bits for that.

Student: You might need to give 2 bits for that; whether is 3 operand 2 operands or a single operand.

So, you keep 2 bits for that that.

Student: We can use 0 0 0 0 to make mention with there is no operands there (Refer Time: 04:28).

So, if you put 2 operands here; that means, see this are actually needed for 3 operand instruction these are all needed if I just put 2 operands here then what will happen only I will have 2 more operands left.

Students: No sir, we can reuse that remaining spaces that.

Remaining is only 2 bits.

Student: no sir if we have 2 operand instructions 2 operand instruction we can use only like 8 bits; the other 4 bits they actually they are use for in addressing a 3 operand can be used for as opcodes in 2 operands.

So how many, so, then tell me you are getting some idea now can you tell me how many 3 address operand 2 address operands instructions how many of each you can have. So, the moment I say 2 operands are used to specify the number of operand 2 bits are used to specify the number of operands; I could have maximum 4 here, right; no if I want 3

address operand instructions with 3 address operands then 12 bits are used for that addressing, remaining you have 4 bits in which you say that 2 bits will always be used to indicate how many operands are there 0 1 2. So, I will have remaining 2 more operands by that I could have 4 instructions.

Student: If the opcode is 0 0 0 4 0s if the opcode is 4 0s then its 0.

So, essentially this is what he says is not that optimal its correct, but that is not optimal now con you start your solution

Student: If you are using 3.

Student: Then you have 12 opcodes.

12 opcodes; 12 bits;

Student: 12 different operations.

So, if I have if I have 3 yeah there.

Student: 4 bits left.

Yeah there will be 4 bits left here yes.

Student: If there are first 4 bits the value is 0 then we can assume it to be 0 address if it is one it will be one address if it is 2 it will 2 address otherwise it will represent opcode.

So, you mean to say I will have 13; 13 instructions.

Student: 13 different operations.

So, you will now you mean some 13 for this.

Student: Similarly if it is 0 or one or 2 then we can use some next 4 bits for the opcode and other 2 will be operands.

So, how do you how do you go about this how will you form a decode.

Student: First we can evaluate that value of first 4 bits.

So, I have to find out whether it is 0 one or 2 right and still you get only 13 bit correct now let me give you a solution. So, if it is; if the first 4 bits first that is most significant. So, this is the most significant bit this is the least significant bit you have to write first 4 Msbs; non 0 right the first 4 Msbs are non 0 if you take the value of the first 4 Msbs then it is say 3 address operand 3 address instructions. So, so I will have everything. So, here all things are possible from 0 0 0 0 0 0 1 to 1 1 1 1 if any of these combinations are there then it is say 3 address operand if I have all 0s then it is a it is not a 3 address operand. So, by this I will have 15 such opcodes, now if this is 0 0 0 0 then I will go to a next op one this is not a 3 address code.

So, it could be a 2 address code or anything if this is 0 0 0 0 if this is not 0 0 0 0 if this is not 0 0 0 0 then it is a 2 address opcode. So, what is 2 address first 8 Msbs sorry second sorry, sorry, sorry, sorry, sorry second 4 Msbs non 0 plus first 4 Msbs 0 and plus means and and first 4 Msbs 0 this is. So, how many I will have here I will have 15 opcodes correct I will have 15 of this now the next one address is third 4 Ms third 4 Msbs non 0 and first and second 4 Msbs 0. So, I could have 15 of these.

And 0 addresses all first second and third 4 most significant bits or 0s. So, everything is 0 then this is 16 0 addresses. So, I could totally have 45 plus 16 61 instructions are you able to follow right. So, I will check the first 4 m s b if it is non 0 then it is going to be. So, if it is non 0 it is going to be a 4 address instruction. So, how do you do how do you find out whether 4 bits are non 0 what is the easiest way of finding whether 4 bits are non 0 r.

You just do an r operation this output of this r will tell me whether it is a 3 address instruction or not if I take the end of this I too I put the next 4 Msbs here right and if I take the end of it, it is going to tell me what it is going to tell ne whether it is a 2 3 address instruction or not. So, this will tell me sorry 2 address instruction this will tell me 3 address this will tell me 2 address if I take the next 4 bits and do it here this will give me whether it is a one address instruction.

Student: Op 4 supposed to be 0 sir then we have to take not and;

What is it?

Student: for 2 operand cases, the first portion is (Refer Time: 12:35) 0.

No, it is not; it is.

Student: There is a not and then why first 4 should be 0; sir and will also report 0 then.

Yeah if it is 0 that is what if it is;

Student: Gap meaning 0 there and one in the;

If it is 0 then it is not a 3 I am saying it is not a 3 with this oh you are saying. So, if it is one it is going to be 3 address if it is 0 it is not 3 address. So, by this I can find out whether 3 address or not now if this is one it is going to be 2 address no, no, I need to put a not and say yeah, yeah, some circuitry here I will leave it to you.

But essentially it will be that whole decoding will be subject to some 2 to 3 levels of gate max can be done much faster. So, this is one very interesting thing that we need to keep up when we are trying to encode the instruction one of the important thing in architecture is to find the list of instruction then how to encode them. So, I say add what is the equivalent 0s and ones for that add that is what we call as encoding. So, the instruction encoding itself is should be done very carefully.

So, that you know my decoding becomes easy. So, there were multiple suggestions that came here, but each of them had their issues it will not be as simple as what we are trying to do here at the same time when I am doing the encoding I should also keep in mind that I should have enough space for maximum number of instructions when we design a small processor like this 16 bit instruction with this I may not have sixty one instruction to start with I may have some 25 30 instructions, but tomorrow when we go and make it to 61 instruction.

Suppose I do an encoding exclusively for that 25 or 30 then what will happen tomorrow I want to add instructions as the architecture grows then it becomes extremely complex. So, when we do the encoding it should also because if you take the evolution of all the long standing instruction set architectures if you look at 80-85 versus 80-86 to 83-86 to the current Intel processor x86 manual set you go and see the number of instructions have grown over a period of time.

And for me to grow the number of instructions I basically need a hook to grow it if I am not going to do something systematic like this right then we may land up with a instruction that is instruction set encoding it is extremely complex. So, this is one of the very important thing that is why we just take this concept this concept is called as expanding opcodes you can just check the web for this there are many materials on this expanding opcodes the book that talks about this is tanenbaum; tanenbaums structured computer organization the book covers this in some detail, but this is the detail max detail that you need to know are you able to follow.

So, once you arrive at an instruction set you say these are all the types of instruction these are all the actions I want to take these are all the type of operands. So, many general purpose registers are needed all these suggestions you come after you come to that then we need to do what we call as encoding and when we do this encoding we need to be sort of systematic. So, that might my decoding becomes easy right you followed this are you able to appreciate this fact.

So, with this I think we have done quite a bit of thing on instruction set architecture we will move on to the next stage which is pipelining. So, now, we will go and look at the

modern current processors what we call as super scalar architectures. So, which can do more than one instruction per cycle it has lot of concurrency we will see all those things in great detail in the in the next 2 to 3 weeks right.

Student: I have a doubt sir clique size limited here right, but in case of 2 operands we can actually have more than 16 operands that is used currently.

Yeah you can, but I have given them these are the assumptions we are taken out first itself I told you operands mean only 4 bits for addressing I made that assumption.

Student: Up to 3 address; it is completely utilised, but then for 2 address we are not using.

No, no, if 2 3 if I do not have 16 bits I cannot put get this sixty one instructions of constant length right then your decoding becomes extremely complex the moment I say I will have variable size instructions then your decoding becomes extremely complex that is the that is the reason we are driving towards risk in some sense any other doubts.