A Pronunciation System for Binary Numbers

Matthew C. Bartsh

Abstract

The author propose a pronunciation system for binary numbers. The system is optimized for ease of learning, since an easily learned algorithm generates the names. The system is secondarily optimized for speed of pronunciation, using very short names for the binary numbers.

Why current pronunciation systems for binary are inadequate for the math classroom.

Let's dive in using fourteen as an example.

Fourteen, when converted from our everyday base ten to a binary number is written '1110'. Sometimes a subscript 'two' or subscript '2' is appended making it '1110two' or '1110_2'. Note that this seems to be best that Medium.com can do to display a subscript 'two' or '2'. I will use '1110two' as being the safer, less ambiguous format.

Most math teachers pronounce 1110two 'one one one oh base two', or 'one one one zero base two', or if the context makes it unambiguous, just 'one one one oh', or 'one one one zero'. This works, but it is not ideal because one is simply reading out the digits. It is like pronouncing '27' as 'two seven' rather than 'twenty-seven'. Or pronouncing 7^2 as 'seven superscript two' rather than 'seven squared'. Or pronouncing French 'chat' (which is pronounced in French roughly as 'shah' and means 'cat'—the meowing animal) as 'see aitch ay tee'.

Not many math teachers, thank goodness, would pronounce 14 ÷ 2 as 'one four divided by two'. But when teaching arithmetic in binary, the teacher probably would pronounce the exact same problem, 1110two ÷ 10two as 'one one one zero divided by one zero'. This seems to be a great pity. The reason for it may be the lack of a well-known way to pronounce binary numbers that is analogous to the way we pronounce base ten numbers when doing arithmetic or counting.

I say 'when doing arithmetic or counting' because when the number is merely a phone number or other code or data, and therefore not a number that that represents a quantity, and not a number whose magnitude needs to be understood, and not a number that will be used in any calculation, then it is fine and very efficient to recite the string of numerals.

You could probably find a math teacher who 'pronounces' 1110two as 'fourteen'. This also works, but it is not ideal because it is not so much pronouncing the binary number as calculating the decimal representation and pronouncing *that.* It is like 'pronouncing' 7^2 as 'forty-nine', or 'pronouncing' French '*chat*' as 'cat', which would be calculating what 7^2 is and pronouncing *that*, and translating French '*chat*' into English 'cat' and pronouncing *that.*

Perhaps half of all people that have an opinion on it, if the Internet ‘answers’ sites are to go by, think that 1110two should be pronounced ‘one thousand one hundred ten base two’. This is a terrible method, because that way you can never safely shorten it to just ‘one thousand one hundred ten’ because of the danger of it being understood as being the familiar one thousand one hundred ten (of base ten). Also, there will *always* be the danger that you will be understood to mean 10001010110two which is 1110ten converted to a binary format. There is no way to make the listener understand that you don’t mean the latter, no matter how you phrase it. Whether you say ‘one thousand one hundred in/written in/read as/base two’ it will never be clear that you don’t mean 10001010110two. It is like pronouncing French ‘chat’ as English ‘chat’ (as ‘chit-chat’).

If you demonstrate or teach counting, arithmetic, and measuring in binary while reciting the numbers like phone numbers, you will have a false comparison with decimal unless you pronounce the decimal numbers in the same way.

It is for the above reasons that I propose that math teachers, and indeed anyone who wishes to appreciate arithmetic, counting, measurement, and other mathematical thinking using binary numbers should pronounce binary numbers in a way that is analogous to how we pronounce our everyday base ten numbers *when not reciting a phone number or a room number or other code or data.*

If you demonstrate or teach counting, arithmetic, and measuring in binary while reciting the numbers like phone numbers, you will have a false comparison with decimal unless you pronounce the decimal numbers in the same way.

The student has little chance of appreciating the full beauty and power of binary when all numbers are reduced to strings of numerals pronounced like binary telephone numbers. And with decimal counting using normal good decimal pronunciation being compared with binary counting using mere recitation of numerals the student cannot make an accurate or enlightened comparison between the two bases.

Counting aloud in binary using my pronunciation system.

Try counting from one to sixteen in binary, pronouncing the binary numbers in a way that is roughly analogous to how you’d do it in base ten:

1two, pronounced ‘one’. Note that 1two is analogous to 1ten because each is the smallest number and the smallest numeral in its base, but also to 9ten because each is the biggest single digit number and the biggest numeral in its base.

10two, pronounced ‘two’. Note that 10two is analogous to 10ten because each is the smallest two digit number.

11two, pronounced ‘two one’. Note that 11two is analogous 11ten because it is one more than 10two but also to 19ten because each is the largest two digit number that starts with a ‘1’ and also to 99ten because each is the largest two digit number.

100two, pronounced ‘four’. Note that 100two is analogous to 100ten because each is the smallest three digit number but also analogous to 900ten because each is the biggest three digit number that ends in ‘00’.

101two, pronounced ‘four one’. Note that 101two is analogous to 101ten because each is one bigger than the smallest three digit number in its base. This is not the only analogy here.

110two, pronounced ‘four two’.

‘111two, pronounced ‘four two one’. Note that 111two is analogous to 999ten because each is the largest three digit number in its base.

1001two, pronounced ‘eight one’.

1010two, pronounced ‘eight two’.

1011two, pronounced ‘eight two one’.

1100two, pronounced ‘eight four’.

1101two, pronounced ‘eight four one’.

1110two, pronounced ‘eight four two’.

1111two, pronounced ‘eight four two one’.

10000two, pronounced…what? ‘Sixteen’ would seem to be the logically implied name. But ‘sixteen’ is a base ten number. It’s short for ‘six and ten’. That’s confusing. And it will be worse when we get to bigger numbers. ‘Thirty-two’ (‘three tens and two’), ‘sixty-four’ (‘six tens and four’), and ‘one hundred twenty eight’(ten tens and two tens and eight’, and so on, getting longer and longer and more and more difficult and confusing and more and more obviously being base ten numbers.

The solution is to find short, easily pronounced and easily recalled names to replace ‘sixteen’ for 10000two, ‘thirty-two’ for 100000two, and so on. But there are an awful lot of names needed, because binary numbers are so long. Compare 1000two with 8ten. It’s four digits compared with one. So these names should be super easy to recall, because so many are needed and no one will be able to face learning so many names unless they are super duper easy to learn.

The easiest to recall names would be names that can be *deduced* from the number, and I have created exactly that. This way, no names need to be memorized, only the way to deduce the name from the number needs be memorized. All that needs to be memorized is the algorithm which is equivalent in difficulty to learning to count to ten in a very easy foreign language.

But for now, to keep things simple, I’ll just tell you that my new name for 10000two is ‘ri’ and my new name for 100000two is ‘li’, and my new name for 1000000two is ‘shi’. 10000000two is ‘ki’. 100000000two is ‘fi’. 1000000000two is ‘pi’.

It would be easier to read and write the binary numbers if there were commas or spaces, so here they are with commas every three digits, like we do with base ten. And for good measure, I’ll show you the names that aren’t really needed, but exist as part of the system, for the numbers 1two, 10two, 100two, and 1000two.

1two is ‘si’. 1ten. Optional.

10two is ‘ti’. 2ten. Optional.

100two is ‘ni’. 4ten. Optional.

1,000two is ‘mi’. 8ten. Optional.

10,000two is ‘ri’. 16ten.

100,000two is ‘li’. 32ten.

1,000,000two is ‘shi’. 64ten.

10,000,000two is ‘ki’. 128ten.

100,000,000two is ‘fi’. 256ten.

1,000,000,000two is ‘pi’ (or ‘hpi’ with a silent ‘h’ added to distinguish it from the word that means the ratio of the diameter of a circle to its circumference). 512ten.

10,000,000,000two is ‘tis’. 1,024ten.

First let’s compare my names with the three rival sets of names it is needs to beat out.

Would you prefer to say ‘tis’ (rhyming with ‘miss’) or ‘one zero zero zero zero zero zero zero zero zero zero’? Which would prefer to write or type (thank goodness for the copypaste function)? Which would you prefer to hear?

Would you prefer to say ‘tis’ or ‘one thousand twenty-four’? Once you know it, ‘tis’ is easier, but more importantly, ‘tis’ is the pronununciation of a binary number, while ‘one thousand twenty-four’ is the pronunciation of the decimal number that it translates to.

Would you prefer to say ‘tis’ or ‘ten billion’? The commas being every three digits make it *easy* to say the latter, but in binary there are usually no commas, or if there are, they may well be every two, four, or some other number of digits. And anyway, it’s a nonsense. It’s equal to one thousand twenty four and not ten billion.

Now let’s count in binary from one to li using my new names.

1two, pronounced ‘one’. Note that 1two is analogous to 1ten because each is the smallest number and the smallest numeral in its base, but also to 9ten because each is the biggest single digit number and the biggest numeral in its base.

10two, pronounced ‘two’. Note that 10two is analogous to 10ten because each is the smallest two digit number.

11two, pronounced ‘two one’. Note that 11two is analogous 11ten because it is one more than 10two but also to 19ten because each is the largest two digit number that starts with a ‘1’ and also to 99ten because each is the largest two digit number.

100two, pronounced ‘four’. Note that 100two is analogous to 100ten because each is the smallest three digit number but also analogous to 900ten because each is the biggest three digit number that ends in ‘00’.

101two, pronounced ‘four one’. Note that 101two is analogous to 101ten because each is one bigger than the smallest three digit number in its base. This is not the only analogy here.

110two, pronounced ‘four two’.

‘111two, pronounced ‘four two one’. Note that 111two is analogous to 999ten because each is the largest three digit number in its base.

1,000two, pronounced ‘eight’. Note that 1,000two is analogous to 1,000ten because each is the smallest four digit number in its base.

1,001two, pronounced ‘eight one’.

1,010two, pronounced ‘eight two’.

1,011two, pronounced ‘eight two one’.

1,100two, pronounced ‘eight four’.

1,101two, pronounced ‘eight four one’.

1,110two, pronounced ‘eight four two’.

1,111two, pronounced ‘eight four two one’.

10,000two, pronounced ‘ri’.

10,001two, pronounced ‘ri one’.

10,010two, pronounced ‘ri two’.

10,011two, pronounced ‘ri two one’.

10,100two, pronounced ‘ri four’.

10,101two, pronounced ‘ri four one’.

10,110two, pronounced ‘ri four two’.

10,111two, pronounced ‘ri four two one’.

11,000two, pronounced ‘ri eight’.

11,001two, pronounced ‘ri eight one’. Thank goodness(again) for the copypaste function.

11,010two, pronounced ‘ri eight two’.

11,011two, pronounced ‘ri eight two one’.

11,100two, pronounced ‘ri eight four’.

11,101two, pronounced ‘ri eight four one’

11,110two, pronounced ‘ri eight four two’.

11,111two, pronounced ‘ri eight four two one’.

100,000two, pronounced ‘li’.

Now let’s look at some binary arithmetic.

10two +10two = 100two, pronounced ‘two plus two equals four’.

1,010two +101two = 1,111two, pronounced ‘eight two plus four one equals eight four two one’.

1,010two - 101two = 101two, pronounced ‘eight two minus four one equals four one’.

1,010two ÷101two = 10two, pronounced ‘eight two divided by four one equals two’.

1,010two × 101two = 110010two, pronounced ‘eight two times four one equals li ri two’.

The new names only variant.

If you want to get as far as possible from base ten, using my system, or to make it clear which base you are using without saying it, you should use only the new names, including the optional ones: ‘si’, ‘ti’, ‘ni’, and ‘mi’ instead of ‘one’, ‘two’, ‘four’, and ‘eight’, respectively.

It is conceivable that some students may confuse knowledge about binary with knowledge about decimal. For example, it is conceivable that a student that learns that four plus two is six in base ten may get confused when told that in binary, four plus two is four two. In fact, there is no reason for confusion as four two means four plus two, which is of course six. But *if* a student is confused, or claims to be, one solution would be to solve this or prevent it by using the new names only variant, so that the student learn not that in binary four plus two is four two but instead that ni plus ti is ni ti. Note that there is then no need to say 'in binary' or even explain what binary is, or how it is different from base ten. The student can simply learn it like a new language, as he or she would when hanging out with some relatives that sometimes speak in another language. That the French grandparents sometimes call their cat 'chat' can't possibly cause any confusion, and if the student observes that often enough will soon learn to understand and be able to do the same thing, all without any confusion, and probably with some enhancement of brain function, psychologists tell us.

In the new-names-only system the binary arithmetic examples that I showed earlier are the same except for the guide to pronunciation:

10two +10two = 100two, pronounced 'ti plus ti equals ni'.

1,010two +101two = 1,111two, pronounced 'mi ti plus ni si equals mi ni ti si'.

1,010two—101two = 101two, pronounced 'mi ti minus ni si equals ni si'.

1,010two ÷101two = 10two, pronounced 'mi ti divided by ni si equals ti'.

1,010two × 101two = 110010two, pronounced 'eight two times four one equals li ri two'. 'mi ti times ni si equals li ri ti'.

In the new name only system, counting from one (si) to li, becomes:

1two, pronounced 'si'. Note that 1two is analogous to 1ten because each is the smallest number and the smallest numeral in its base, but also to 9ten because each is the biggest single digit number and the biggest numeral in its base.

10two, pronounced 'ti'. Note that 10two is analogous to 10ten because each is the smallest two digit number.

11two, pronounced 'ti si'. Note that 11two is analogous 11ten because it is one more than 10two but also to 19ten because each is the largest two digit number that starts with a '1' and also to 99ten because each is the largest two digit number.

100two, pronounced 'ni'. Note that 100two is analogous to 100ten because each is the smallest three digit number but also analogous to 900ten because each is the biggest three digit number that ends in '00'.

101two, pronounced 'ni si'. Note that 101two is analogous to 101ten because each is one bigger than the smallest three digit number in its base. This is not the only analogy here.

110two, pronounced 'ni ti'.

'111two, pronounced 'ni ti si'. Note that 111two is analogous to 999ten because each is the largest three digit number in its base.

1,000two, pronounced 'mi'. Note that 1,000two is analogous to 1,000ten because each is the smallest four digit number in its base.

1,001two, pronounced 'mi si'.

1,010two, pronounced ‘mi ti’.

1,011two, pronounced ‘mi ti si’.

1,100two, pronounced ‘mi ni’.

1,101two, pronounced ‘mi ni si’.

1,110two, pronounced ‘mi ni ti’.

1,111two, pronounced ‘mi ni ti si’.

10,000two, pronounced ‘ri’.

10,001two, pronounced ‘ri si’.

10,010two, pronounced ‘ri ti’.

10,011two, pronounced ‘ri ti si

10,100two, pronounced ‘ri ni’.

10,101two, pronounced ‘ri ni si’.

10,110two, pronounced ‘ri ni ti’.

10,111two, pronounced ‘ri ni ti si’.

11,000two, pronounced ‘ri mi’.

11,001two, pronounced ‘ri mi si’.

11,010two, pronounced ‘ri mi ti’.

11,011two, pronounced ‘ri mi ti si’.

11,100two, pronounced ‘ri mi ni’.

11,101two, pronounced ‘ri mi ni si’

11,110two, pronounced ‘ri mi ni ti’.

11,111two, pronounced ‘ri mi ni ti si’.

100,000two, pronounced ‘li’.

How the infinite set of names were generated, and how they can be deduced from the numbers and vice versa.

It’s time to share the rules that allow the names of each binary number to be deduced. Note that the math teacher does not need to share these rules with the students, because the latter do not need to know them to count and do arithmetic in binary using the names.

Here is the full set of rules for mapping the consonants to the ten numerals:

0 = s, 1 = t, 2 = n, 3 = m, 4 = r, 5 = l, 6 = sh, 7 = k, 8 = f, 9 = p.

(The choice of specific mappings is due my modifying in a consistent way a well-known set of mappings used in memorizing numbers using something called the Major System of mnemonics https://en.wikipedia.org/wiki/Mnemonic_major_system)

Also, once the students know the ten names of the powers of two from ^{20} to ^{29} (which are si, ti, ni, mi, ri, li, shi, ki, fi, and pi) the rules for the mapping of the ten numerals to consonants and vice versa can be deduced from those ten names. So perhaps instead of teaching the rules, one could teach the first ten names of powers of two. If you want to make that into a single chunk of knowledge, why not teach the students to pronounce 1,111,111,111two. It is pronounced, “pi fi ki shi li ri mi ni ti si”. It could be learned like a phrase in a foreign language: “pi , fi ki shi, li ri mi, ni ti si”. Note that ni ti si is seven, and li ri mi is fifty-six (seven eights), and fi ki shi is four hundred forty eight (seven sixty-fours).

Note that ‘ti’ is the second name in the list but is two raised to the first power, so it is probably not a good idea to call ‘si’ the first power of two.

Thus, since ‘i’ means ‘two to the power of’ and ‘p’ means ‘9’ (the base ten numeral) ‘pi’ means 2^9. Likewise ‘pip’ means 2^{99}. And ‘pipip’ means 2^{999}.

Every power of two is specified. You just add the letter ‘i’ as many times as is needed as padding to generate words that are pronounceable and mean ‘two to the power of (something)’ while the superscript part of the power (the something) is specified by the permutation of the consonants. Two to the power one hundred one (2^{101}) is ‘tisit’, pronounced ‘tissit’. Two the power two hundred fifty-three (2^{253}) is ‘nilim’, pronounced ‘nillim’.

An infinite number of pronounceable names are specified by my system.

The system is optimized for easy recall, or deduction if recall is not possible, of the name of any power of two when the number is thought of and vice versa.

Uses for my pronunciation system in the classroom.

A simple way to get students familiar with binary numbers is for the teacher to dictate (or play a recording of) some binary numbers while the students write them down. For example, the teacher says, ‘ eight four two one’ or ‘mi ni ti si’ and the student writes down, ‘1,111’ (or ‘1,111two’ if it needs to be distinct from decimal).

Or the teacher could dictate ‘two one times two one’ or ‘ti si times ti si’ and the student writes down, ’11 × 11’ and then solves it and writes down, ‘1,001’ and perhaps writes the answer in longhand as, ‘mi si’.

A final comparison with other methods of pronouncing binary numbers.

By the way, imagine what a mess it would be if the teacher dictated, using that method that is so often voted up on the Internet answer sites when binary is asked about, ‘eleven times eleven’, and the student wrote down the answer in longhand as, ‘one thousand one’. The student is liable to forget in which base eleven time eleven equals one hundred twenty-one. Bear in mind that the student may well also be studying base eight and/or base sixteen and/or base twelve and/or base six, and presumably pronouncing ‘11’ as eleven in all of them.

The other popular method of pronunciation: ‘one one times one one equals one zero zero one’ works but it’s not as good, because it isn’t clear what quantities each one stands for.

And consider what happens when the dictated numbers are bigger. 49,248ten is 1,100,000,001,100,000two which I would pronounce as “til tir shi li”. The two rival methods mentioned just now would pronounce it, “one quadrillion one hundred trillion one million one hundred thousand”, and “one one zero zero zero zero zero zero one one zero zero zero zero zero”.

How concise is my system?

Very concise. 1,111,111,111two pronounced as a string of numerals would be 'one one one one one one one one one one binary'. Thirteen syllables including 'binary'). With my system, which needs no 'binary' to disambiguate the base in order to disambiguate the number, it is 'pi fi ki shi li ri mi ni ti si' which is ten syllables. Three syllables fewer. And this is the least flattering example. Replace any one of those one's with a zero and you need to add a syllable (if it will be pronounced 'zero') to the telephone number style recitation of the numerals, while subtracting a syllable from my pronunciation of it.

So in fact it is not so much more efficient of computer scientists to recite the numerals as *easier*. Saying 'one one one one' is not *quicker* than saying than saying 'mi ni ti si' if you are fluent in my system. In fact the latter is probably quicker and clearer, as long as the listener is also reasonably fluent. This is because the syllables are by design short, high pitched, phonologically. The use of phonologically 'unvoiced' consonants as far as possible: 's' not 'z', 't' not 'd', and so on, is also intended to make the names clearer phonologically as well as more regular (constrained) and therefore easier to learn.

Binary fractions.

In decimal we pronounce 12.19ten as 'twelve point one nine'.

The analogous system for binary would be to pronounce 1001.1101two as 'eight one point one one zero one' or if the new names only format is preferred, 'mi si point si si zero si'.

Unfortunate names generated by the algorithm.

2^{11} is/would be 'tit', which is not too upsetting because it's the name of a type of bird, but 2^{61} is/would be 'sh*t', and 2^{90} is/would be 'pis'. 2^{61} definitely needs a new name. Some possible substitutes would be: shith, shid, chit, chid, and chith. I think that ideally an algorithm would determine how unfortunate names would be changed. Which names are unfortunate enough to need changing is subjective and ever changing as slang words are coined or drop out of use, and so perhaps the best thing at this stage is just to let the algorithm do its thing, but make the system customizable by users, and let users decide which names are too unfortunate. For example, a teacher who is using the system in school will probably want 'sh*t' changed to something else, but maybe someone using it in a video on his or her own website wouldn't care. Fortunately they are rare, about one percent to three percent of the powers of two.

Further reading.

What you have seen in this article is just the tip of an iceberg. To keep things simple, I have limited myself to discussing binary. But my infinite set of names for the powers of two is sufficient for any base that is based on a power of two. That means octal, hexadecimal, as well as base four, base thirty-two, base sixty-four, and so on. Infinitely many bases, each with an infinite number of powers of two, each one with a name taken from binary.

Also, systems modeled on the one described here can be used to generate an infinite set of names for base twelve, that can supplement or replace the names for the powers of twelve currently used in base twelve : 'dozen' and 'gross' are two of them being 12^1 and 12^2 respectively and written 10twelve and 100twelve respectively, and analogously base one hundred forty-four, base one thousand seven hundred twenty-eight, and every other power of twelve.

To see a recent article of mine (that is even easier to understand than this one) about how my names for the powers of two can be reused in (and are sufficient for) octal in my pronunciation system for octal that is also analogous to base ten and therefore well suited to the math classroom click this link:

https://bartshmatthew.medium.com/how-octal-numbers-should-be-pronounced-by-math-teachers-72ddb42e8e03?source=your_stories_page-------------------------------------

so no new names are needed for octal click this link.

To go much deeper and see an older rough around the edges article that includes more or less all my ideas on this topic, click the link below.

https://bartshmatthew.medium.com/a-super-easy-way-to-pronounce-binary-numbers-and-numbers-in-some-other-number-bases-4778efb1ef03

By Matthew Christopher Bartsh on March 15, 2021.