



Mixed Signal Circuits

Intimately combined analog and digital signal processing





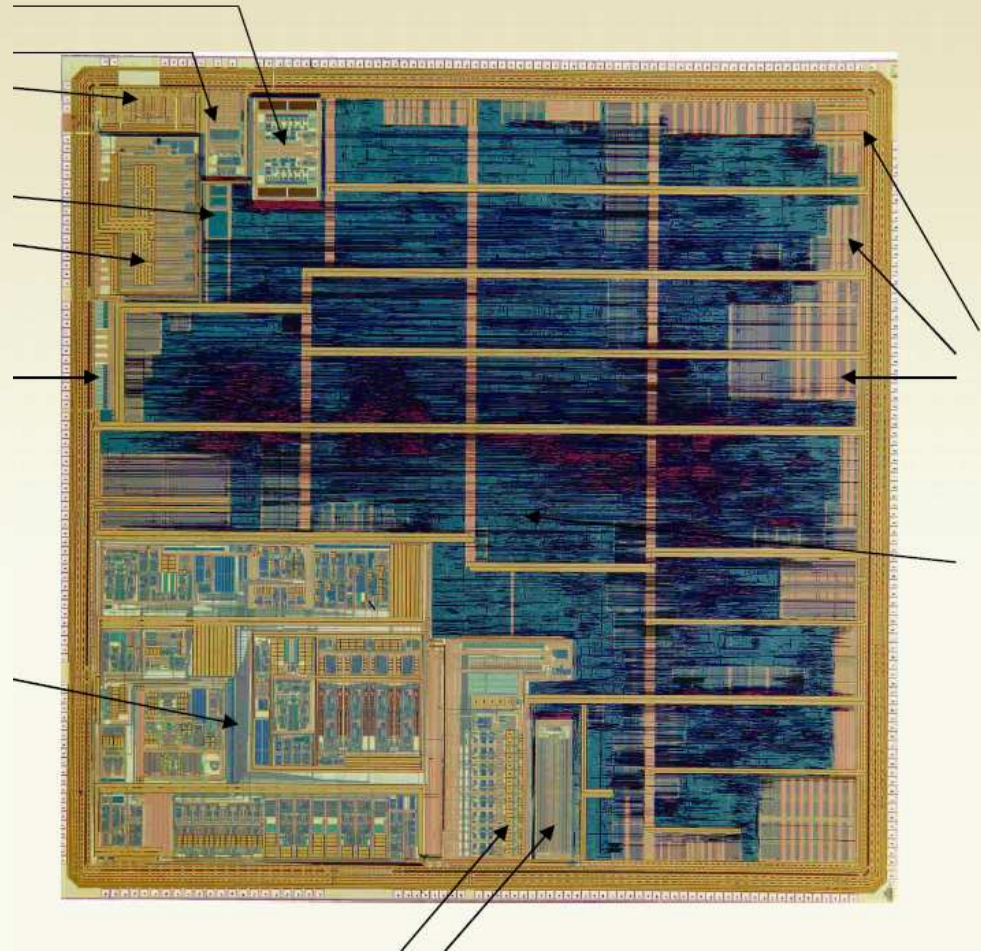
A Mixed Signal Chip

Look at this chip: do you see the analog blocks and the digital blocks?

Today's mixed signal circuits are systems that are made of analog cells and digital cells on the same chip. That's what we mean by "mixed signal" chips.

We probably have "analog designers" and "digital designers" in the company working together on a common project.

This seems the obvious way to build complex systems on a chip.



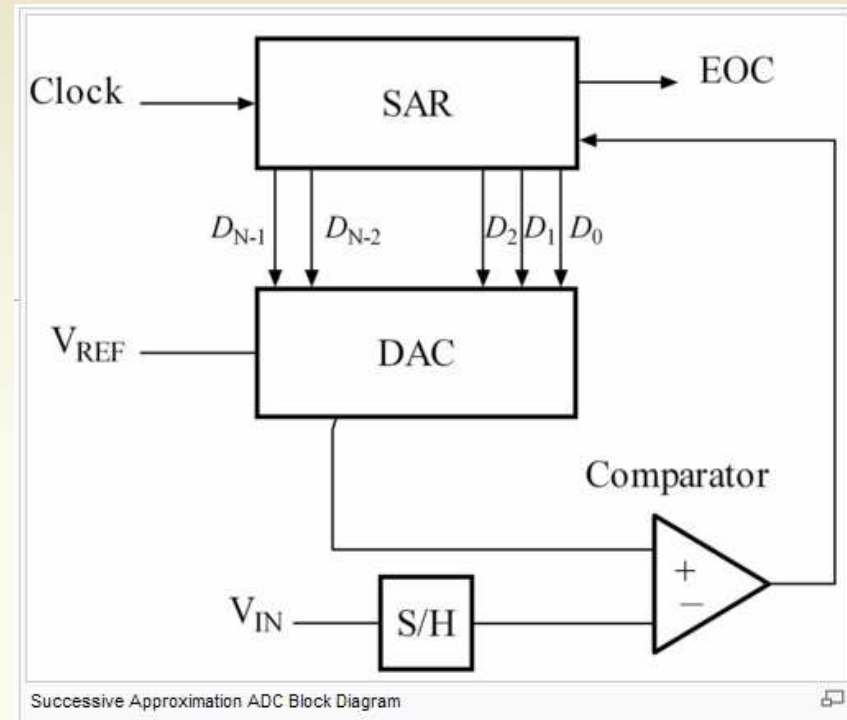
ESS Martin Mallinson
CMOS ET 2008





A Mixed Signal Device

- Arguably the first mixed signal device: the humble SAR ADC*
- Is this any different from our Mixed Signal Chip?
- Actually, yes it is – it differs in one key respect...



*Successive Approximation (Register) Analog to Digital Converter



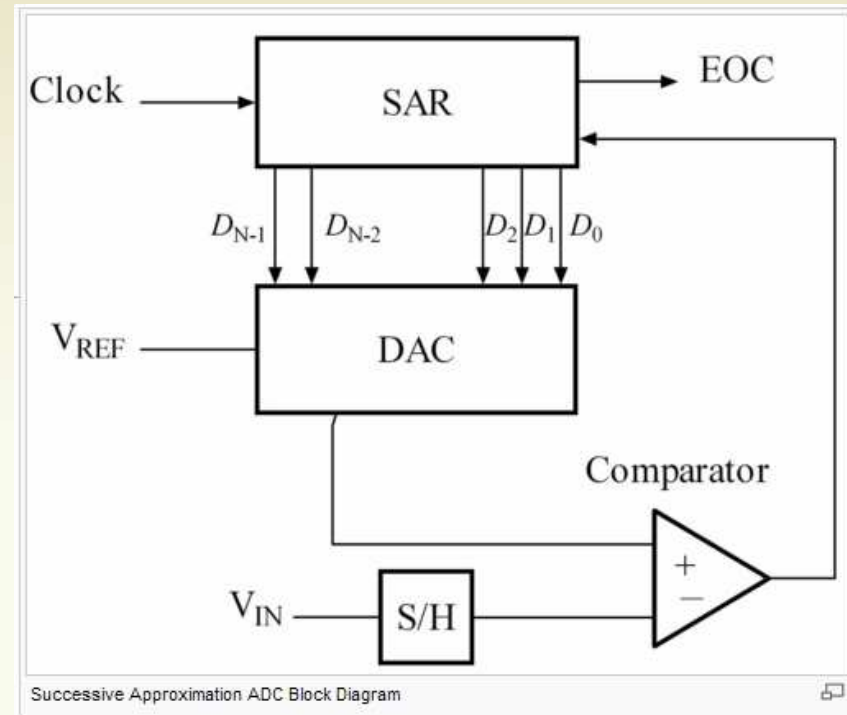
ESS Martin Mallinson
CMOS ET 2008





SAR ADC

- ... take away the digital and the functionality ceases
- ... take away the analog and the functionality ceases (Obviously)
- In this simple design the digital and the analog are intimately connected:
The desired functionality emerges out of the combination of the two.





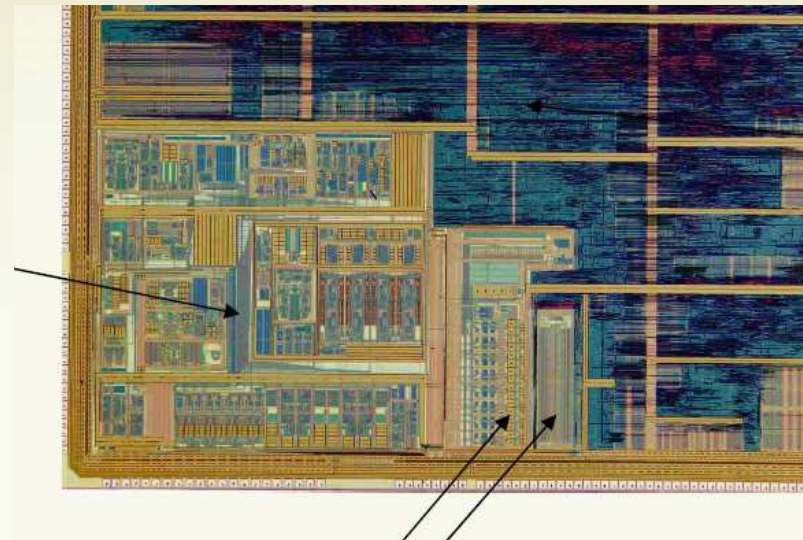
Are these intimately connected?

The bottom left is actually called the AFE – the Analog Front End – it's amplifiers, filters and comparators continue to work even if the digital is shut off.

It is not intimately connected to the digital circuit.

But why is this observation even interesting?

So what if it is not intimately connected? – it still works...



ESS Martin Mallinson
CMOS ET 2008





Better Performance is possible

- Intimately connected mixed circuits can have performance exceeding the sum of their parts...
- To convince you of this I am going to show two examples:
 - An ADC that does not require a complex AAF* because it shares some key aspects between the analog and the digital circuit.
 - A replacement for a PLL that needs no VCO or control loop and yet generates any arbitrarily specified frequency by exploiting a simple analog and a simple digital block – neither of which is much use without the other.
 - There are many more such circuits waiting to be invented...

*Anti Aliasing Filter



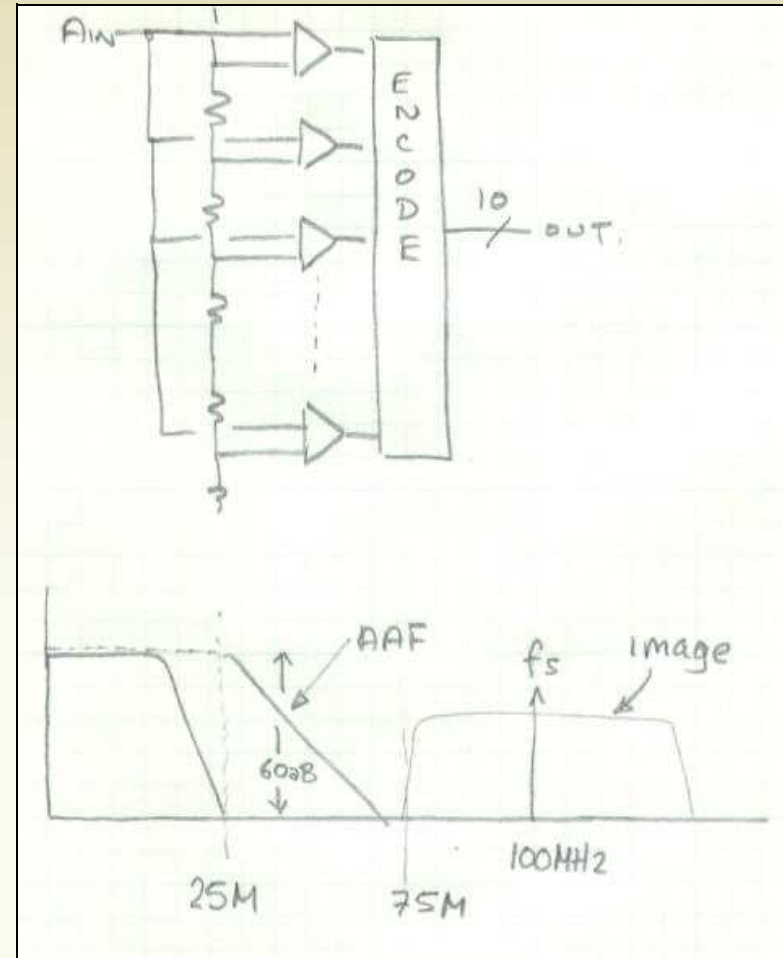
ESS Martin Mallinson
CMOS ET 2008





Flash ADC Example (1)

- A Flash ADC is conceptually simple: it has a comparator for every level.
 - On modern CMOS processes such ADC's are relatively small and practical.
- Actually, the AAF is more difficult:
 - Example: 25Mhz BW sampled at 10 bits 100Mhz => AAF is >6th order*.



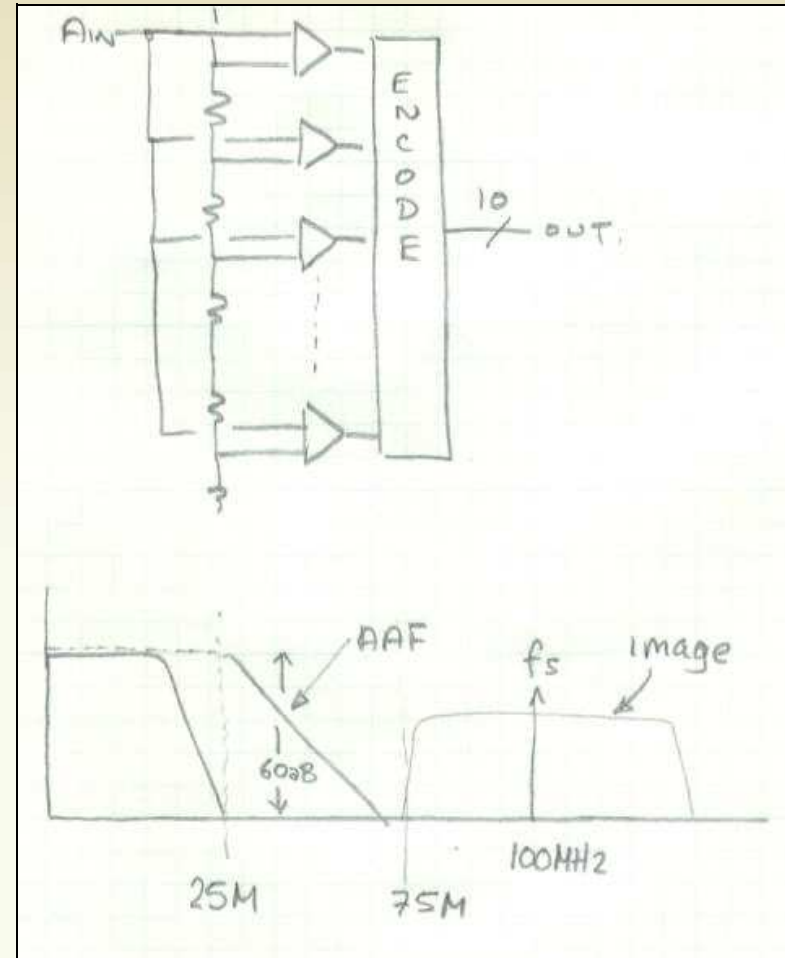
* Using simple approximation of $\log_3(60\text{dB}) = 6.28$



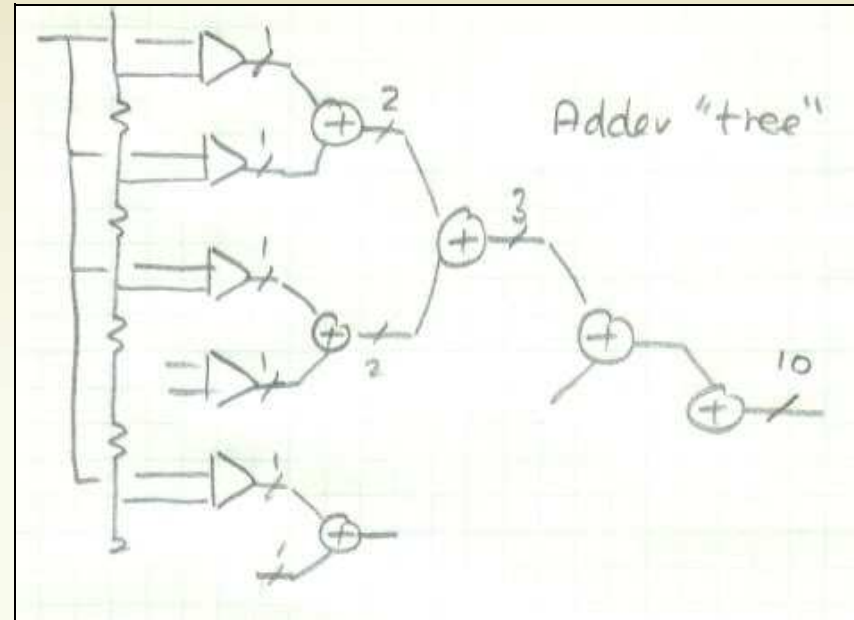


Flash ADC Example (2)

- Do we really need an AAF?
 - Sampling theorem says “of course we do! If we sample at 100Mhz we can’t tell the difference between 90Mhz and 10Mhz – both create 10Mhz in the digital domain”.
 - Remarkably, we will find that the above is correct, but it does not imply that we need a sophisticated AAF. The solution lies in the digital domain...



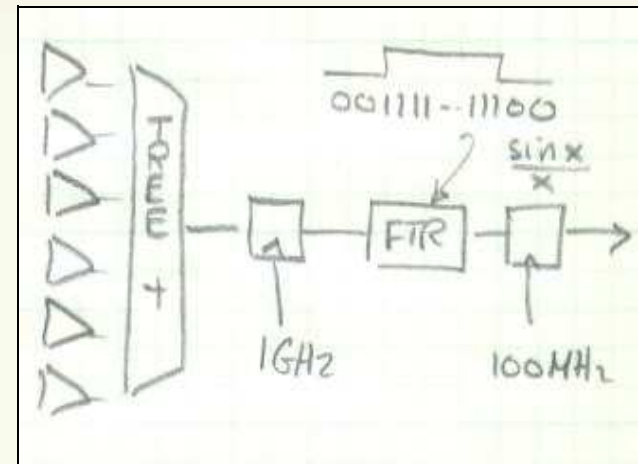
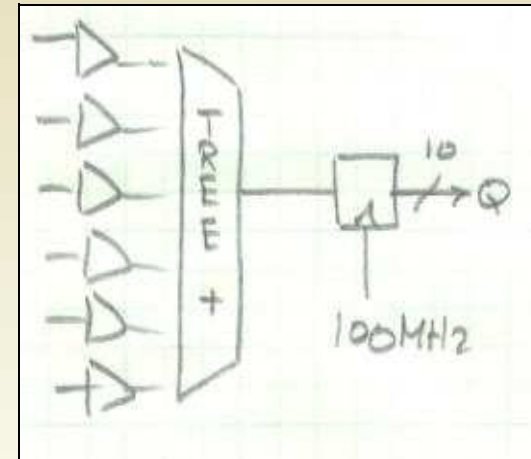
- How does that encoder work?
 - This is the “best” way: an adder tree – this makes and ‘nth order’ bubble correction and is even fault tolerant to some degree..





Flash ADC Example (4)

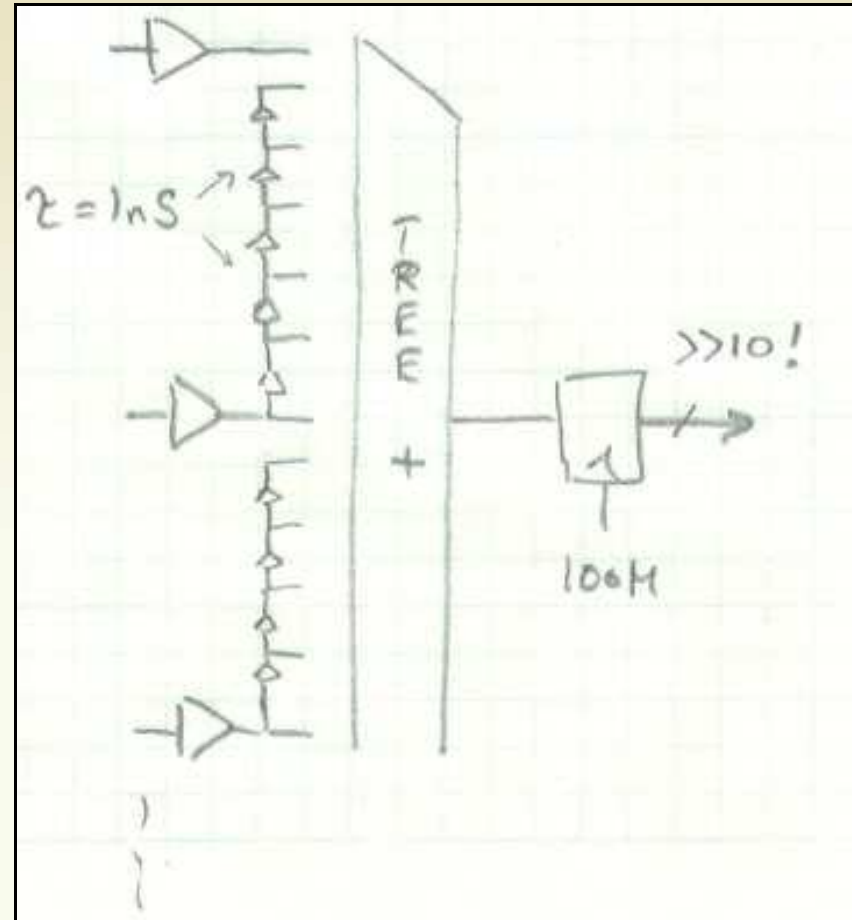
- Lets imagine that tree in one block and note that the act of sampling occurs in the digital domain, not the analog domain.
- Of course then if we could sample at 1Ghz and digitally filter the result we would much reduce our AAF – we would then need to remove the alias not at 75MHz, but at 975Mhz – a second order filter would do it.
- But this is no help at all – we just asked for a 1Ghz ADC – of course it got easier!
- But wait...isn't filtering a linear operation? Can it be put before the adder tree? And, a simple $\sin x/x$ filter, well isn't that just another adder tree? ...





Flash ADC Example (5)

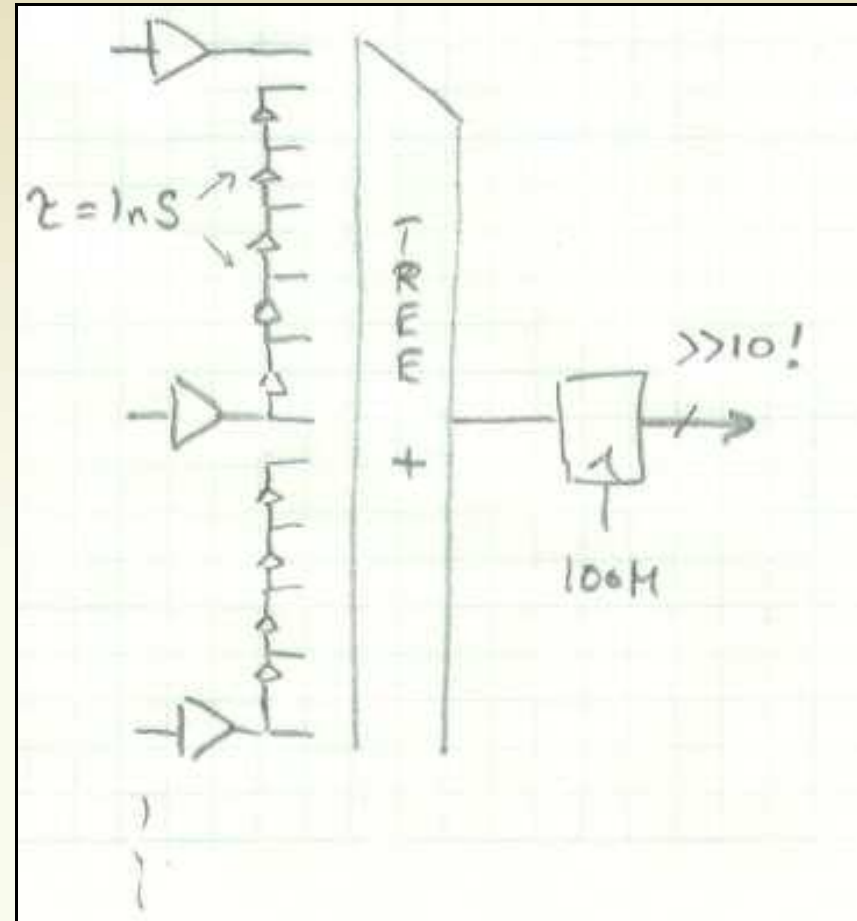
- All those assumptions on the prior page are actually true – we can do this:
 - We simply add a series of, for example, 10 delay elements of 1nS to every one of the comparator outputs. Then we extend the adder tree to incorporate those extra inputs.
 - This is exactly the same as the system diagram on the bottom of the last page – from an anti-aliasing point of view this is now a 1Ghz ADC with a “box car” response (i.e. $\text{sinc}(x)/x$) of 10 elements equivalently clocked at 1Ghz. And yet, the logic clock is only 100Mhz.





Flash ADC Example (6)

- There is more:
 - The output resolution is much higher – we may not need 1024 comparators any more
 - The AAF for our example goes from 7th to 3rd order.
 - The “tree” can implement a higher order “Pascal” filter*
 - All this logic amounts to 97k gates. That’s small – much smaller than the equivalent 7th order filter.
 - (The way to think about this is that the circuit now has deep and wide notches at the first 10 multiples of the clock. That’s why the AAF is much easier)

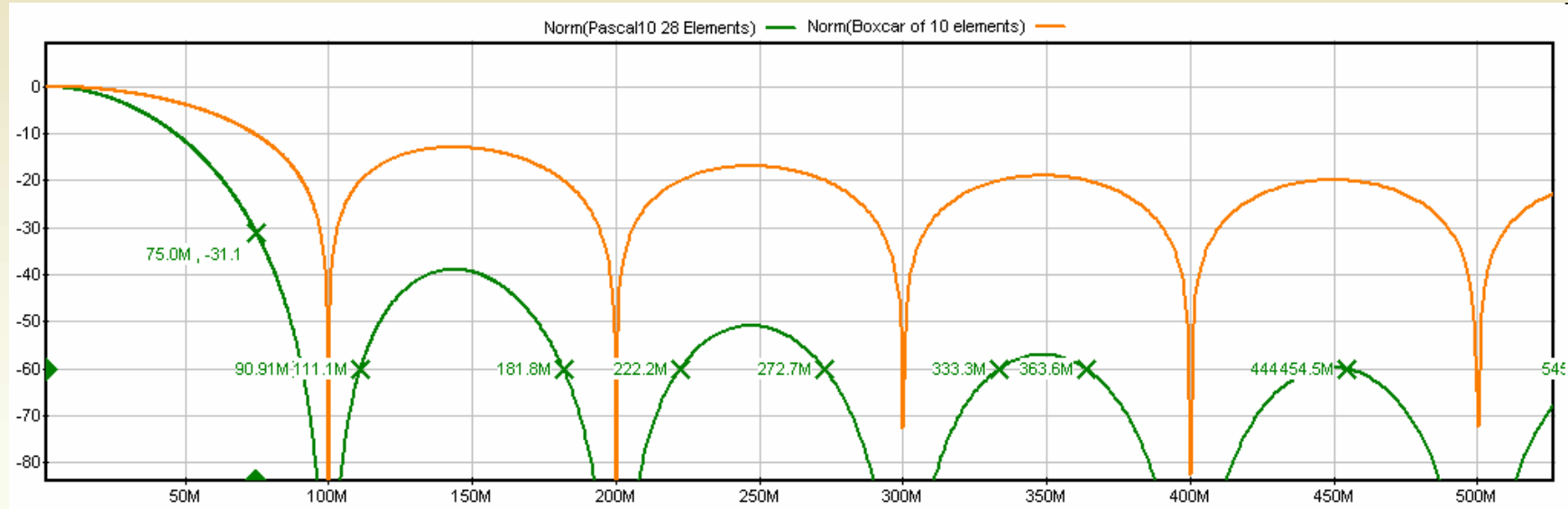


* This is beyond the scope of this presentation, but such a filter extends the width of the zeros that fall on the clock multiples. (It is interesting because it also can be made in the adder tree with no multipliers or complex logic).





Flash ADC Example (7)



Potential alias around the clock is removed – $\pm 10\text{MHz}$ is -60db , 25MHz is -30db .
Note how the “pascal10 28” solution far exceeds the sinc/x (this is the one in 97k gates).



ESS Martin Mallinson
CMOS ET 2008

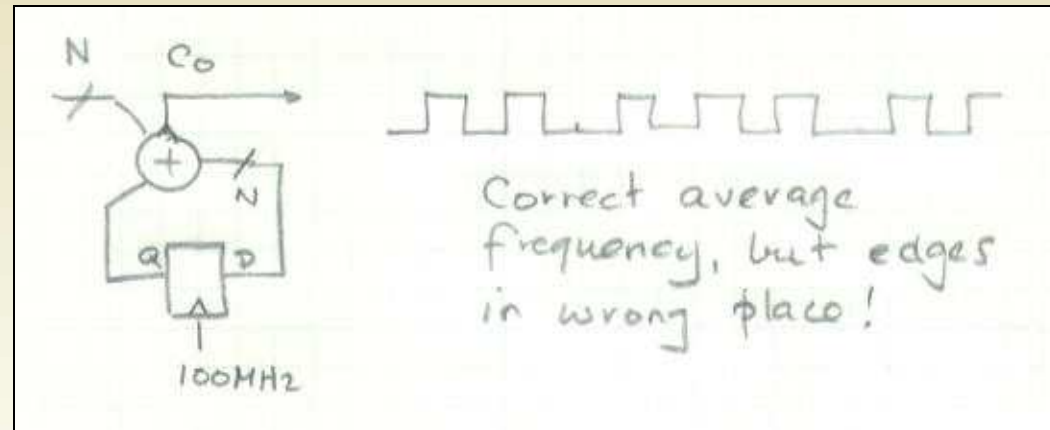




Frequency Generator (1)

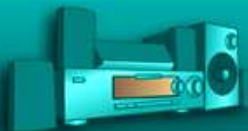
This simple digital circuit makes any frequency to any accuracy: it is well known.

Its problem is that the frequency may well be correct, *on average*, but the jitter is horrendous – all edges are actually on the clock edges!



This is a modulo arithmetic adder. Interestingly, if you observe the adder output at the moment of overflow, that residue is the timing error*... we can use this fact.

*Gerard Wischermann shows this in US patent [5.195.044](#)



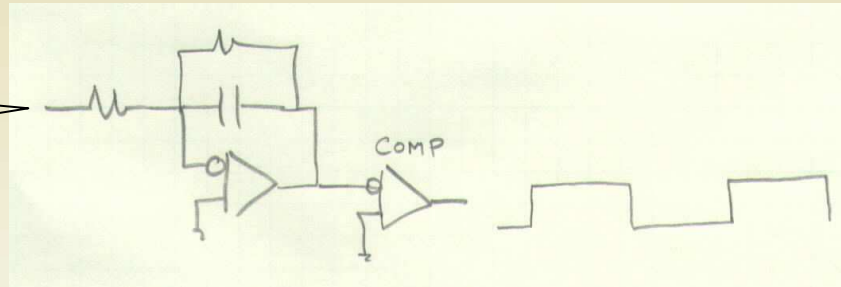
ESS Martin Mallinson
CMOS ET 2008



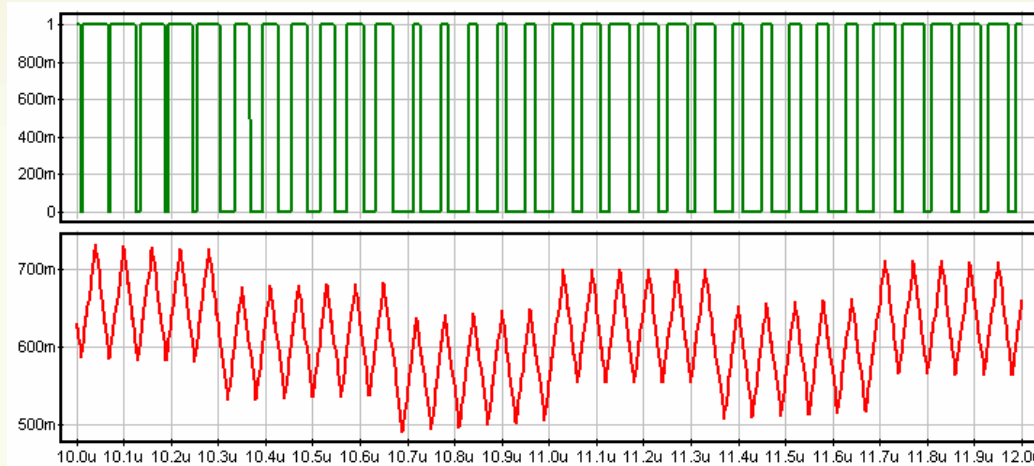


Frequency Generator (2)

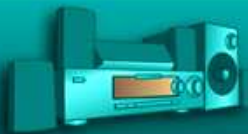
Carry-out of previous circuit



We can apply the carry-out signal that has the correct average value to an integrator like this:



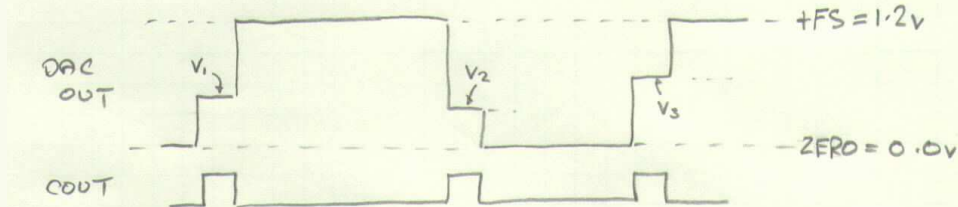
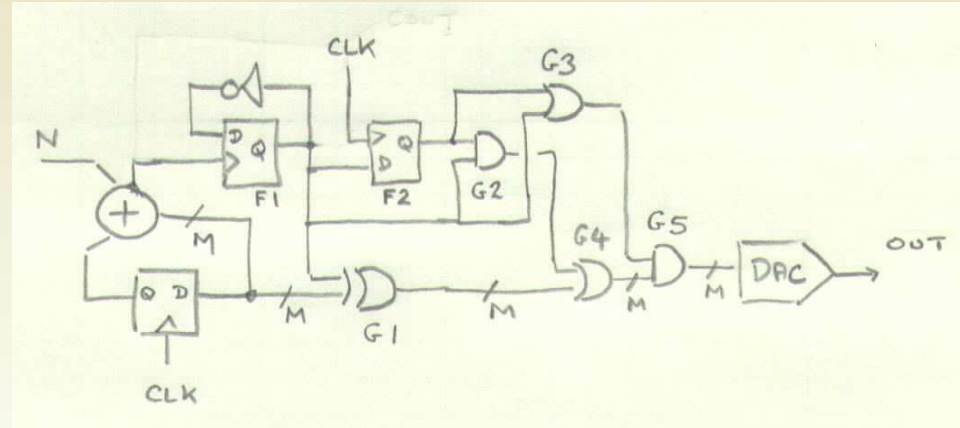
It does not work very well: the green is the output signal, the red is the signal on the integrator output. We have achieved nothing: the jitter is still bad, but we can begin to see why: it is because the red signal has a changing average value.



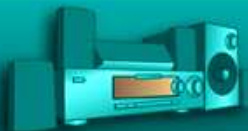


Frequency Generator (3)

This look a bit tricky doesn't it? It's just logic, but that DAC has appeared. That DAC causes the output to look like this: it does not go from zero to full-scale directly, it spends just one clock cycle at an intermediate value.



What does this achieve?

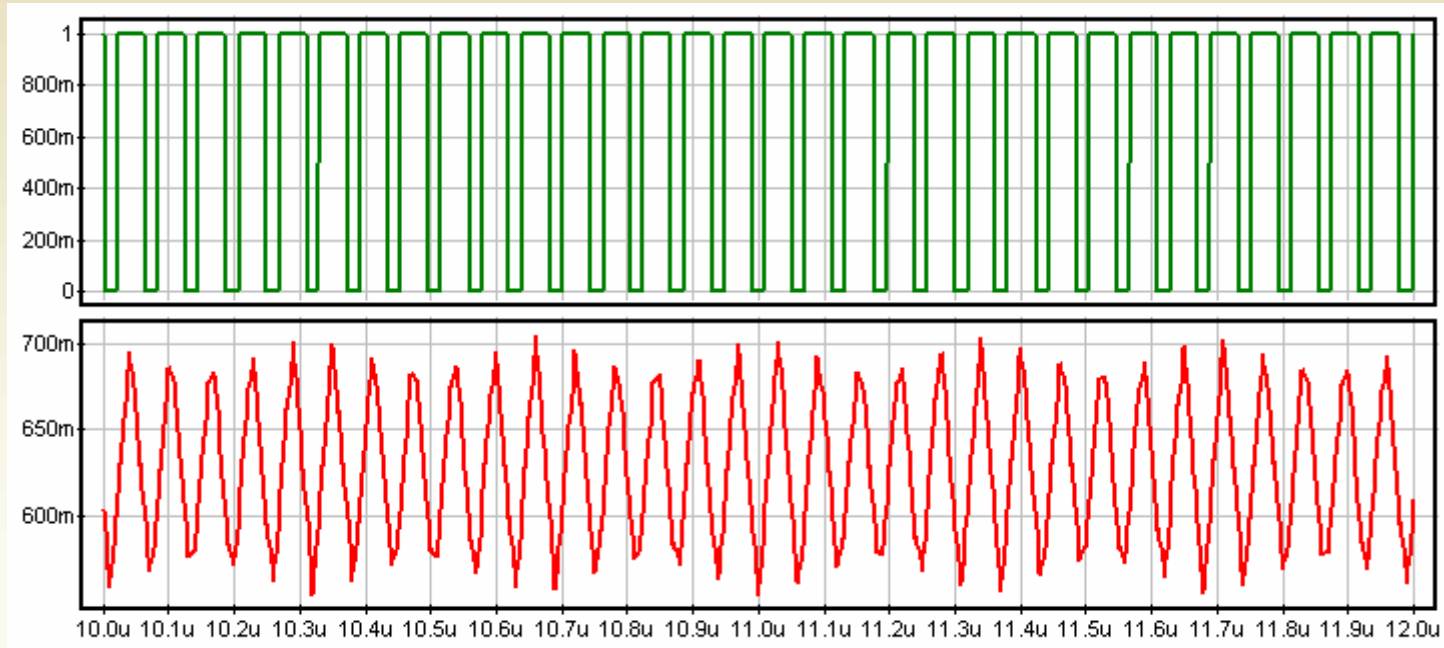


ESS Martin Mallinson
CMOS ET 2008





Frequency Generator (4)



The average value changes have gone – the edges have moved to the correct place – *between the input clock pulses* – the jitter just dropped from 10nS to about 25pS!

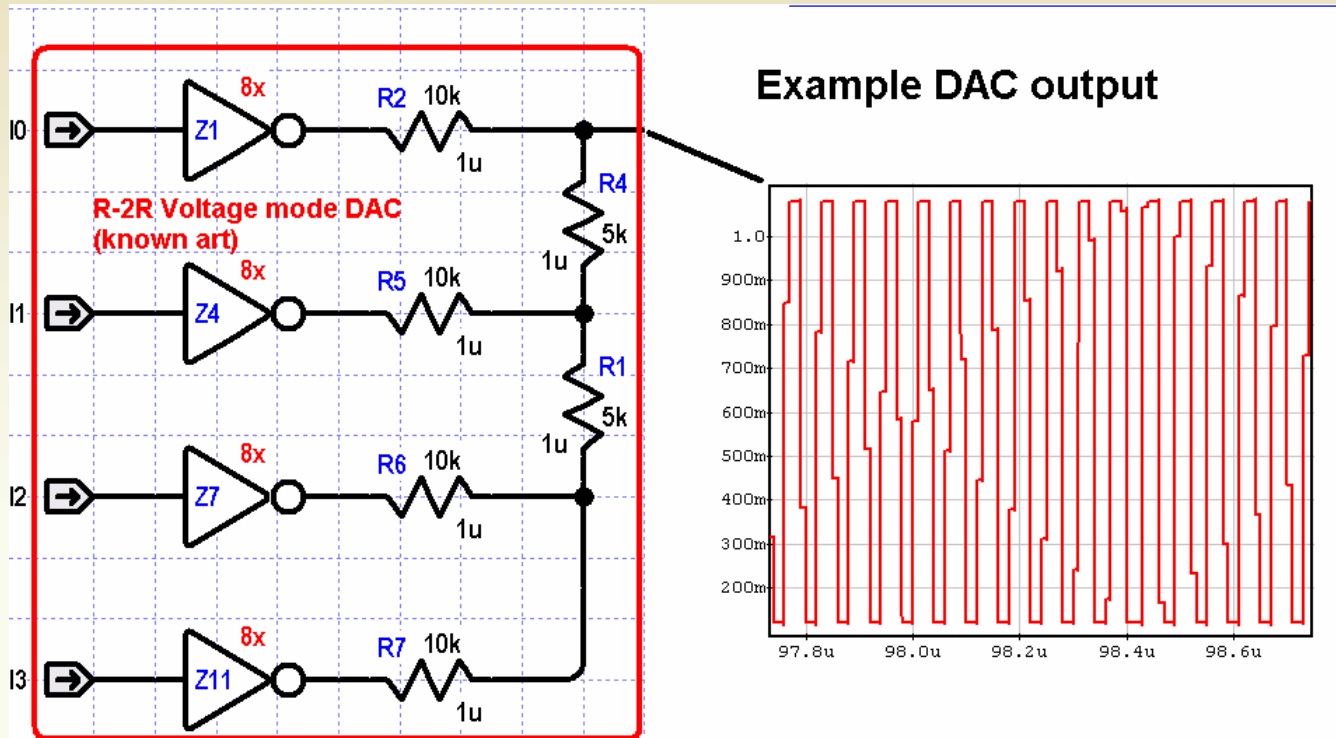


ESS Martin Mallinson
CMOS ET 2008





Frequency Generator (5)



That "DAC" can be just this.



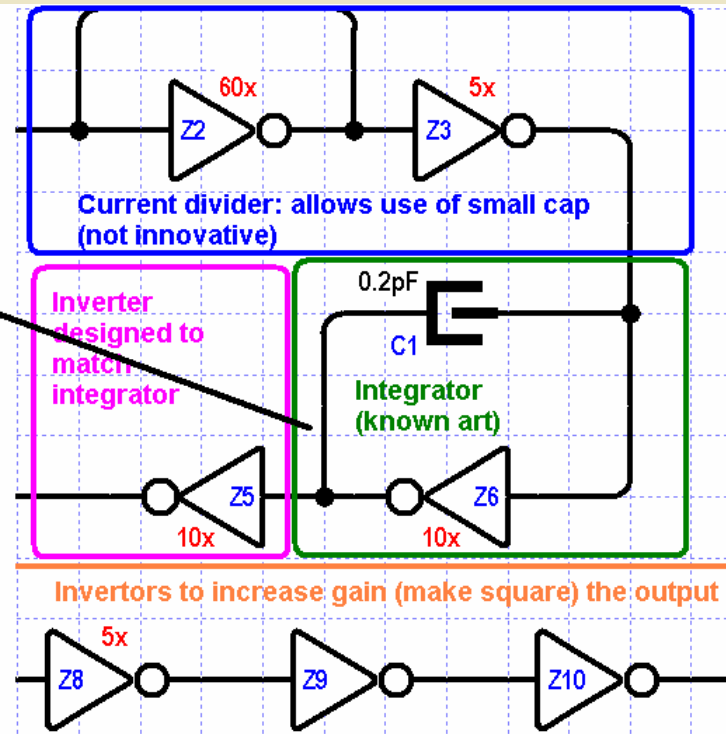
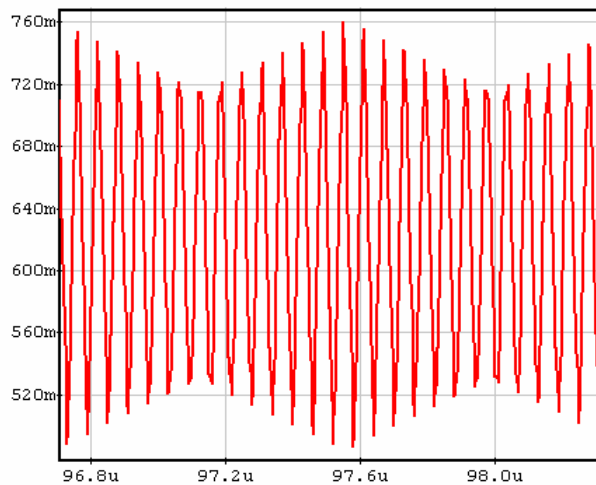
ESS Martin Mallinson
CMOS ET 2008





Frequency Generator (7)

This is the signal on the integrator output



That integrator can be just this.



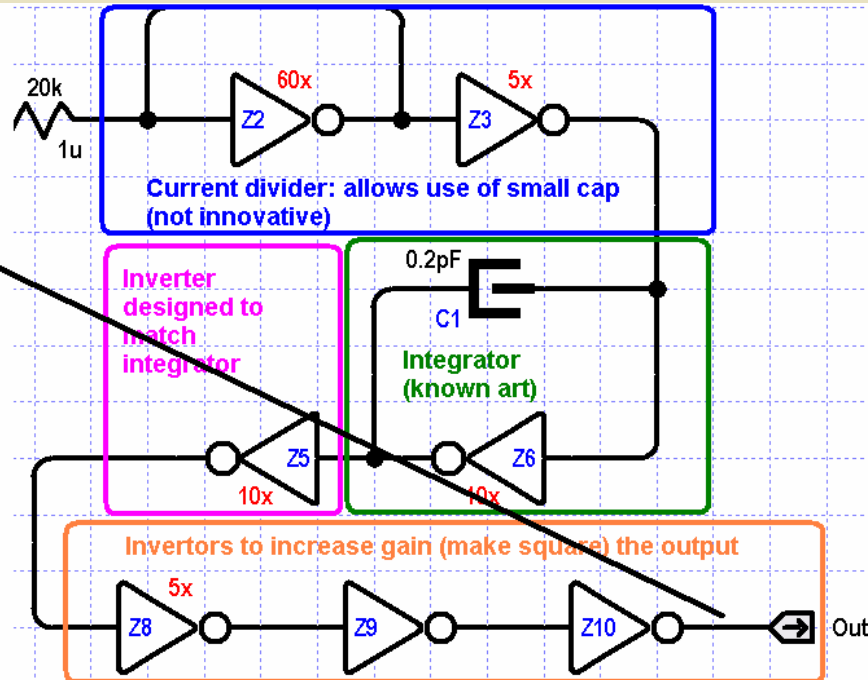
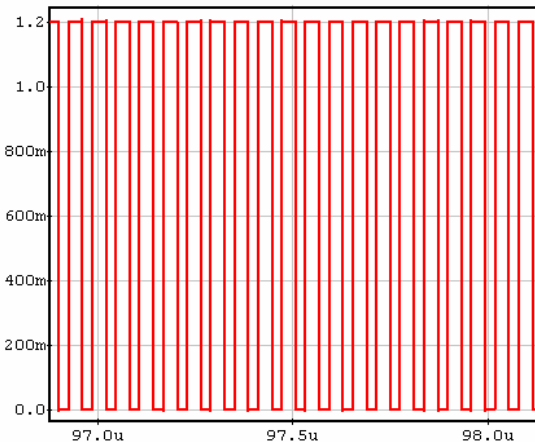
ESS Martin Mallinson
CMOS ET 2008





Frequency Generator (8)

Final output: jitter 22pS RMS



The output circuit is just this. Basically, the entire generator is digital CMOS with 6 resistors and a 200fF cap. It's unconditionally stable, it has no control loop and it's frequency resolution can be as high as you need – 32 bits for example.

But I should mention: it's patented by ESS -- ask us if you want to use it.

[Also, to help your analysis: the peak jitter is equal to the clock period divided by the number of DAC levels – in a relatively simplistic example sub-100pS is achievable]



ESS Martin Mallinson
CMOS ET 2008

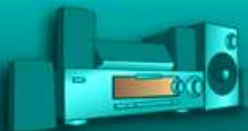




Summary

- I hope you found the two examples interesting.
 - There are many more...
- The examples show that there are two kinds of mixed signal circuits:
 - Those that have digital and analog circuits side by side on the chip
 - These make complex systems and work well, nothing wrong with them, but they do not exploit “mixed signal” design as they could.
 - Those circuits that cannot work independently: they are more than the sum of their parts
 - These are the kind of circuits that have added value in the marketplace
(and they are fascinating and challenging to design and build!)

Martin Mallinson, ESS Technology, Kelowna, BC
CMOSET 2008



ESS Martin Mallinson
CMOSET 2008

