Amar Kumar-CED17I029

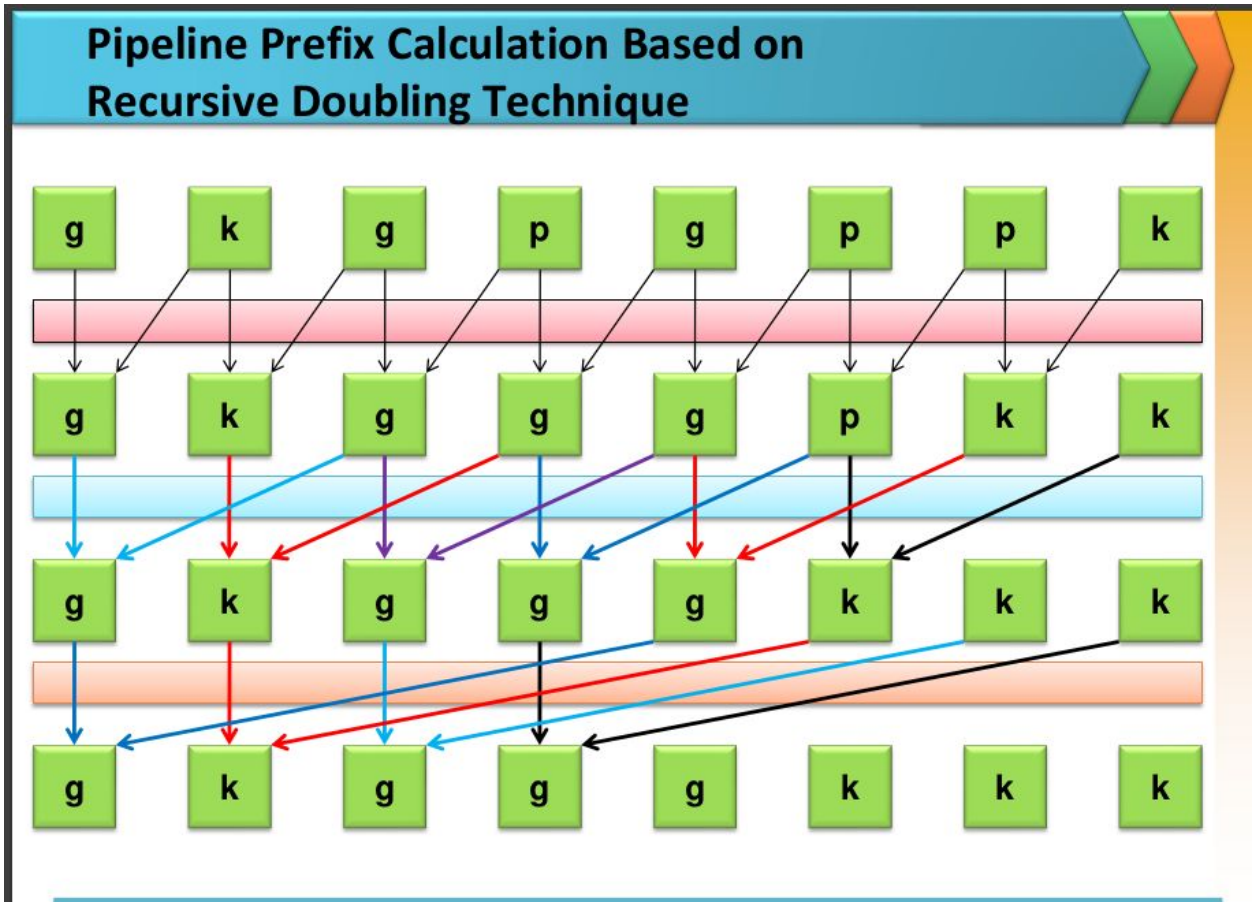
16 bit Recursive Doubling

Introduction

We already know that addition using normal adder takes a lot of time i.e O(n) where n is number of bits of input because normal adder adds bit by bit and wait for carry from previous bit to proceed to next bit. To overcome this recursive doubling is introduced which does addition in log(n).

Pipelining and logic components

(assume delay of the gates as following:- NOT-1D, AND-1D, OR-1D, XOR-3D)
In recursive doubling, we have total of 7 stages to perform. So in order to get pipelined version of recursive doubling we need to pipeline these 7 stages.
Stage 1 - we calculate p,k,g which are propagate, kill, generate respectively. We get kill when we add 0 with 0, we get propagate when we add 0 with 1 or 1 with 0 and we get generate when we add 1 with 1. This is done using 15 AND gates and 15 OR gates which will run in parallel. So it has a delay of 1D
Stage 2 - in this stage, we make kmap by ourself to generate formula for calculation of partial sum and carry. After formula generation we calculate partial sum and carry with the help of 2 AND gate in parallel and 1 OR gate. So this has a delay of 2D
Stage 3 to stage6 - it is same as the stage 2. Only difference is which should bit should be AND with which bit. It can be understood with the help of diagram
Stage 7 - In this stage, we get final sum. For this we use 2 XOR gates for calculating sum of each bit. We calculate sum of all bits in parallel. So this has a delay of 2*2D = 4D



This image is just for reference for 8 bits number

Comparison of pipelined and non-pipelined architecture

Let us assume that we have 1000 instructions

In pipelined version,
The first instruction will take a delay of 15D.
All other instructions will take a delay of 4D to complete addition
So average delay can be 4D*1000 = 4000D
Accurate delay = 15D*1 + 4D*999 = 4011D

In non-pipelined version,
Total delay for 1000 instructions to get added will be 15D*1000 = 15000D

% of Improvement can be achieved with respect to non-pipelined architecture.

%Improvement using pipeline = $\frac{non-pipelined}{pipelined}\ x\ 100$

%Improvement using pipeline = $\frac{15000}{4000} \; x \; 100 \; = 375\%$