

Exhibit C-12: “Splash 2: FPGAs in a Custom Computing Machine,” D.A. Buell et al., 1996 (“Buell”)

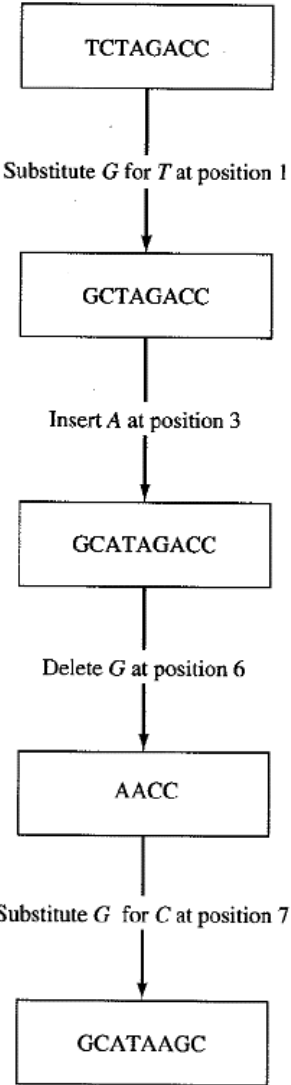
A book chapter entitled, “Splash 2: FPGAs in a custom Computing Machine,” by D.A. Buell, J. M. Arnold, and W. J. Kleinfelder, (“Buell”) was published in 1996, and is therefore prior art to U.S. Patent No. 7,225,324 (“’324 Patent”) at least under 35 U.S.C. §§ 102(a) and (b).

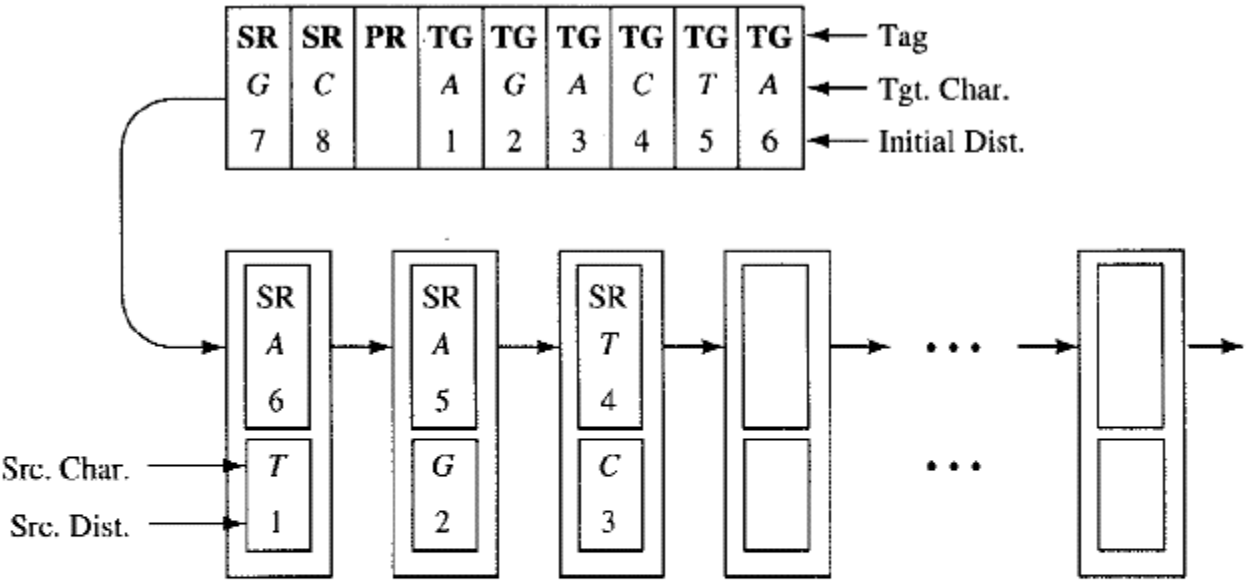
The citations presented herein are exemplary and not exclusive; each prior art reference as a whole discloses each and every limitation of the claims. A citation to a figure or figure reference numeral incorporates by reference the discussion and/or explication of such figure or feature/component referenced by the reference numeral. Further, the mapping in this chart is based on Amazon’s present understanding of Plaintiffs’ interpretation of the asserted claims of the patent-in-suit as reflected in Plaintiffs’ infringement contentions. Nothing in the chart should be regarded as necessarily reflecting how the prior art references would apply to claim elements of the asserted patent under a proper interpretation of the claims. Disclosures cited for dependent claims incorporate by reference the disclosure included herein for the corresponding independent claim.

Asserted Claim of ’324 Patent	Exemplary Disclosure of Buell
<p>[1A] A method for data processing</p>	<p>Buell at 97: “With the onset of the Human Genome Initiative [3] and constant advances in genetic sequencing technology, <i>genetic sequence data</i> are being generated at an ever increasing rate. As a result, biologists are faced with an influx of new sequences that they would like to classify and study by comparing them to <i>existing databases</i>. The analysis of a newly generated sequence typically involves <i>searching the databases</i> for similar sequences. <i>With the enormous size of the databases, fast methods are needed for comparing sequences</i> [11].</p> <p>In this chapter, we describe <i>two systolic array architectures for sequence comparison...</i>”</p> <p>Buell at 100: “The locality of reference shown in Figure 8.3 can be exploited to produce systolic algorithms in which communication is limited to adjacent <i>processors...</i></p> <p>The systolic architecture and <i>data flow</i> shown in Figure 8.5 were used in the design of P-NAC of Lipton and Lopresti [12], a custom VLSI chip for DNA sequence comparison. Each <i>processing</i> element (PE) computes the distances along a particular diagonal of the distance matrix.”</p>

Exhibit C-12: “Splash 2: FPGAs in a Custom Computing Machine,” D.A. Buell et al., 1996 (“Buell”)

Asserted Claim of '324 Patent	Exemplary Disclosure of Buell
	Buell at Figure 8.1:

Asserted Claim of '324 Patent	Exemplary Disclosure of Buell
	<div style="text-align: center;">  <pre> graph TD A[TCTAGACC] -- "Substitute G for T at position 1" --> B[GCTAGACC] B -- "Insert A at position 3" --> C[GCATAGACC] C -- "Delete G at position 6" --> D[AACC] D -- "Substitute G for C at position 7" --> E[GCATAAGC] </pre> </div> <p>FIGURE 8.1 Listing of Operations to Transform <i>TCTAGACC</i> into <i>GCATAAGC</i>. Character matches are assumed to have a cost of 0 and are not shown. Assigning a cost of 2 for a substitution, 1 for deletion, and 1 for insertion, the cost of the transformation is 6.</p> <p>Buell at Figure 8.9 [emphasis added]:</p>

Asserted Claim of '324 Patent	Exemplary Disclosure of Buell
	 <p>The diagram illustrates a Unidirectional Systolic Array. At the top, a sequence of tags and characters is shown: SR, SR, PR, TG, TG, TG, TG, TG, TG. Below these are the source characters G, C, and target characters A, G, A, C, T, A. Below that are source distances 7, 8, and target distances 1, 2, 3, 4, 5, 6. Arrows indicate that the SR tag points to the source character and distance, the PR tag points to the end of the source stream, and the TG tag points to the target character and distance.</p> <p>The array consists of a series of processing elements (PEs) connected in a pipeline. The first PE receives 'Src. Char.' (T) and 'Src. Dist.' (1). Its SR register contains 'A' and its distance register contains '6'. The next PE has SR 'A' and distance '5'. The third PE has SR 'T' and distance '3'. The fourth PE is empty. The fifth PE is empty. The sixth PE is empty. The seventh PE is empty. The eighth PE is empty. The ninth PE is empty. The tenth PE is empty. The eleventh PE is empty. The twelfth PE is empty. The thirteenth PE is empty. The fourteenth PE is empty. The fifteenth PE is empty. The sixteenth PE is empty. The seventeenth PE is empty. The eighteenth PE is empty. The nineteenth PE is empty. The twentieth PE is empty. The twenty-first PE is empty. The twenty-second PE is empty. The twenty-third PE is empty. The twenty-fourth PE is empty. The twenty-fifth PE is empty. The twenty-sixth PE is empty. The twenty-seventh PE is empty. The twenty-eighth PE is empty. The twenty-ninth PE is empty. The thirtieth PE is empty. The thirty-first PE is empty. The thirty-second PE is empty. The thirty-third PE is empty. The thirty-fourth PE is empty. The thirty-fifth PE is empty. The thirty-sixth PE is empty. The thirty-seventh PE is empty. The thirty-eighth PE is empty. The thirty-ninth PE is empty. The fortieth PE is empty. The forty-first PE is empty. The forty-second PE is empty. The forty-third PE is empty. The forty-fourth PE is empty. The forty-fifth PE is empty. The forty-sixth PE is empty. The forty-seventh PE is empty. The forty-eighth PE is empty. The forty-ninth PE is empty. The fiftieth PE is empty. The fifty-first PE is empty. The fifty-second PE is empty. The fifty-third PE is empty. The fifty-fourth PE is empty. The fifty-fifth PE is empty. The fifty-sixth PE is empty. The fifty-seventh PE is empty. The fifty-eighth PE is empty. The fifty-ninth PE is empty. The sixtieth PE is empty. The sixty-first PE is empty. The sixty-second PE is empty. The sixty-third PE is empty. The sixty-fourth PE is empty. The sixty-fifth PE is empty. The sixty-sixth PE is empty. The sixty-seventh PE is empty. The sixty-eighth PE is empty. The sixty-ninth PE is empty. The seventieth PE is empty. The seventy-first PE is empty. The seventy-second PE is empty. The seventy-third PE is empty. The seventy-fourth PE is empty. The seventy-fifth PE is empty. The seventy-sixth PE is empty. The seventy-seventh PE is empty. The seventy-eighth PE is empty. The seventy-ninth PE is empty. The eightieth PE is empty. The eighty-first PE is empty. The eighty-second PE is empty. The eighty-third PE is empty. The eighty-fourth PE is empty. The eighty-fifth PE is empty. The eighty-sixth PE is empty. The eighty-seventh PE is empty. The eighty-eighth PE is empty. The eighty-ninth PE is empty. The ninetieth PE is empty. The ninety-first PE is empty. The ninety-second PE is empty. The ninety-third PE is empty. The ninety-fourth PE is empty. The ninety-fifth PE is empty. The ninety-sixth PE is empty. The ninety-seventh PE is empty. The ninety-eighth PE is empty. The ninety-ninth PE is empty. The one-hundredth PE is empty.</p> <p>FIGURE 8.9 Data Flow through the Unidirectional Systolic Array. The source sequence is <u>first loaded into the array</u>. The target sequences are <u>then streamed through the array</u>. The tag acts as a simple instruction telling each PE how to process the incoming data. The SR tag instructs an empty <u>PE to load the source character and distance from the input stream</u>. The PR tag marks the end of the source stream. The TG tag signals a target character. Multiple source and target sequences can be carried <u>on the input stream for uninterrupted pipelined processing</u>.</p> <p>Buell showing a method for processing data at Figure 8.12:</p>

Asserted Claim of '324 Patent	Exemplary Disclosure of Buell
	<pre> loop if (TAGin = SR) then if (SRCch = ∅) then SRCch ← CHRin CHRout ← ∅ DSTout ← PDSTin else CHRout ← CHRin endif PDSTout ← PDSTin else-if (TAGin = PR) then if (SRCch = ∅) then DSTout ← PDSTin endif PDSTout ← DSTin CHRout ← CHRin else-if (TAGin = TG) then if (SRCch ≠ ∅) and (CHRin ≠ ∅) then DSTout ← min { PDSTout+ψ(SRCch,CHIRin), DSTin+ψ(SRCch,∅), DSTout+ψ(∅,CHRin) } else-if (SRCch = ∅) then DSTout ← DSTin endif PDSTout ← DSTin CHRout ← CHRin endif TAGout ← TAGin endloop </pre> <p>FIGURE 8.12 Code executed by each PE in the unidirectional array</p> <p>Buell showing a method for processing data at Figure 8.7:</p>

Explore Litigation Insights

Docket Alarm provides insights to develop a more informed litigation strategy and the peace of mind of knowing you're on top of things.

Real-Time Litigation Alerts



Keep your litigation team up-to-date with **real-time alerts** and advanced team management tools built for the enterprise, all while greatly reducing PACER spend.

Our comprehensive service means we can handle Federal, State, and Administrative courts across the country.

Advanced Docket Research



With over 230 million records, Docket Alarm's cloud-native docket research platform finds what other services can't. Coverage includes Federal, State, plus PTAB, TTAB, ITC and NLRB decisions, all in one place.

Identify arguments that have been successful in the past with full text, pinpoint searching. Link to case law cited within any court document via Fastcase.

Analytics At Your Fingertips



Learn what happened the last time a particular judge, opposing counsel or company faced cases similar to yours.

Advanced out-of-the-box PTAB and TTAB analytics are always at your fingertips.

API

Docket Alarm offers a powerful API (application programming interface) to developers that want to integrate case filings into their apps.

LAW FIRMS

Build custom dashboards for your attorneys and clients with live data direct from the court.

Automate many repetitive legal tasks like conflict checks, document management, and marketing.

FINANCIAL INSTITUTIONS

Litigation and bankruptcy checks for companies and debtors.

E-DISCOVERY AND LEGAL VENDORS

Sync your system to PACER to automate legal marketing.